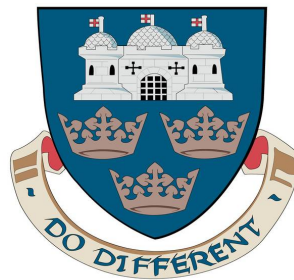


Twitter Mining for Syndromic Surveillance

A THESIS SUBMITTED TO THE SCHOOL OF COMPUTING SCIENCES AT
THE UNIVERSITY OF EAST ANGLIA IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY



ODUWA EDO-OSAGIE
2019

© This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that use of any information derived there from must be in accordance with current UK Copyright Law. In addition, any quotation or extract must include full attribution.

Abstract

Enormous amounts of personalised data is generated daily from social media platforms today. Twitter in particular, generates vast textual streams in real-time, accompanied with personal information. This big social media data offers a potential avenue for inferring public and social patterns. This PhD thesis investigates the use of Twitter data to deliver signals for syndromic surveillance in order to assess its ability to augment existing syndromic surveillance efforts and give a better understanding of symptomatic people who do not seek healthcare advice directly. We focus on a specific syndrome - asthma/difficulty breathing. We seek to develop means of extracting reliable signals from the Twitter signal, to be used for syndromic surveillance purposes. We begin by outlining our data collection and preprocessing methods. However, we observe that even with keyword-based data collection, many of the collected tweets are not relevant because they represent chatter, or talk of awareness instead of an individual suffering a particular condition. In light of this, we set out to identify relevant tweets to collect a strong and reliable signal. We first develop novel features based on the emoji content of Tweets and apply semi-supervised learning techniques to filter Tweets. Next, we investigate the effectiveness of deep learning at this task. We propose a novel classification algorithm based on neural language models, and compare it to existing successful and popular deep learning algorithms. Following this, we go on to propose an attentive bi-directional Recurrent Neural Network architecture for filtering Tweets which also offers additional syndromic surveillance utility by identifying keywords among syndromic Tweets. In doing so, we are not only able to detect alarms, but also have some clues into what the alarm involves. Lastly, we look towards optimizing the Twitter syndromic surveillance pipeline by selecting the best possible keywords to be supplied to the Twitter API. We developed algorithms to intelligently and automatically select keywords such that the quality, in terms of relevance, and quantity of Tweets collected is maximised.

Access Condition and Agreement

Each deposit in UEA Digital Repository is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the Data Collections is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form. You must obtain permission from the copyright holder, usually the author, for any other use. Exceptions only apply where a deposit may be explicitly provided under a stated licence, such as a Creative Commons licence or Open Government licence.

Electronic or print copies may not be offered, whether for sale or otherwise to anyone, unless explicitly stated under a Creative Commons or Open Government license. Unauthorised reproduction, editing or reformatting for resale purposes is explicitly prohibited (except where approved by the copyright holder themselves) and UEA reserves the right to take immediate 'take down' action on behalf of the copyright and/or rights holder if this Access condition of the UEA Digital Repository is breached. Any material in this database has been supplied on the understanding that it is copyright material and that no quotation from the material may be published without proper acknowledgement.

Contents

Abstract	i
1 Introduction	2
1.1 Twitter Mining for Syndromic Surveillance	2
1.2 Research Questions	4
1.3 Research Aims and Objectives	5
1.4 Research Contributions and Outputs	6
1.5 Big Data and Social Media	7
1.6 Syndromic Surveillance	8
1.7 Organisation of Thesis	9
2 Technical Background	11
2.1 Text Mining	11
2.1.1 Text Categorization	12
2.1.2 Sentiment Analysis	12
2.1.3 Information Retrieval	13
2.1.4 Text Clustering	15
2.2 Text Processing: Natural Language Processing	16
2.2.1 Syntactical Analysis	16
2.2.2 Semantic Analysis	19
2.3 Machine Learning for Text Mining	20
2.3.1 Supervised Learning	20
2.3.2 Unsupervised Learning	21
2.3.3 Semi-Supervised Learning	24
2.4 Neural Networks and Deep Learning	26
2.4.1 Recurrent Neural Networks	28
2.4.2 Convolutional Neural Networks	30
2.5 Text Transformation and the Vector Space Model	31
2.5.1 The Term-Document Matrix	32
2.5.2 The Word-Context Matrix	32
2.5.3 The Pair-Pattern Matrix	33

2.6	Learning Neural Representations of Text	33
2.6.1	Word2Vec	34
2.6.2	GloVe	35
2.6.3	Paragraph2Vec	35
2.7	Summary	37

3 A Scoping Review of the use of Twitter for Public Health

Research	38
3.1 Introduction	38
3.2 Method	39
3.2.1 Search Strategy	40
3.2.2 Study Selection	40
3.2.3 Information extraction and analysis plan	41
3.3 Results	42
3.3.1 Study Characteristics	42
3.3.2 Application Domains of Twitter in Public Health . .	45
3.4 Discussion	60

4 Working with Twitter Data: Extraction, Preparation and

Processing	62
4.1	Introduction 62
4.2	The Twitter Application Programmer’s Interface (API) . . . 63
4.3	Data Collection and Preprocessing 64
4.3.1	Location Filtering 65
4.4	Data Cleaning and Preprocessing 67
4.4.1	Retweets 67
4.4.2	Duplicate Tweets 67
4.4.3	URLs 67
4.4.4	Spambots and Articles 68
4.4.5	Labelling 68
4.5	Feature Extraction 69
4.5.1	Word Classes 71
4.5.2	Positive and negative word counts 72
4.5.3	Indicates Asthma Possession 72
4.5.4	Contains “Asthma-Verb” Conjugate: 72
4.5.5	Denotes laughter: 73
4.5.6	Negative emojis/emoticons: 73
4.5.7	Text Embeddings 74
4.6	Summary 77

5 Experimental Methodology: Semi-supervised Classification

for Relevance Filtering	78
5.1 Introduction	78

5.2	Iterative Labelling Algorithms	79
5.2.1	Self-Training	79
5.2.2	Co-Training	80
5.3	Generative Classification Network	81
5.3.1	Architecture	82
5.3.2	Algorithm	84
5.4	Attentive Bi-directional Recurrent Neural Network	85
5.4.1	Word Embeddings	86
5.4.2	Bi-directional Recurrent Neural Network	87
5.4.3	Attention	88
5.4.4	Softmax Layer	89
5.5	Evaluation and Performance Metrics	89
5.5.1	Model Evaluation	89
5.5.2	Feature Evaluation	91
5.5.3	Generalization and Validation	92
5.5.4	Correcting the Class Imbalance	92
5.5.5	Statistical Tests	92
5.6	Summary	93
6	Results of Relevance Filtering and Syndromic Surveillance	95
6.1	Introduction	95
6.2	Feature Analysis	95
6.2.1	Hand-made Features	96
6.2.2	Text Embeddings	98
6.3	Iterative Labelling Experimentation	99
6.3.1	Experiments and Results	99
6.3.2	Discussion	104
6.4	Generative Classification Network Experimentation	106
6.4.1	Experiments and Results	106
6.4.2	Discussion	108
6.5	Attentive Bi-directional Recurrent Neural Network Experimentation	109
6.5.1	Experiments and Results	109
6.5.2	Discussion	111
6.6	Syndromic Surveillance	112
6.6.1	Difficulty Breathing	112
6.6.2	Asthma/Difficulty Breathing/Wheezing	113
6.6.3	Control Syndrome: Diarrhoea	116
6.6.4	Discussion	118
6.7	Summary	122
7	Optimizing the Twitter Syndromic Surveillance Stream: Intelligent and Automatic Keyword Selection for Twitter Stream-	

ing	123
7.1 Introduction	123
7.2 Approaches to Intelligent and Automatic Keyword Selection	124
7.2.1 Similarity Heuristic-Based Keyword Selection	125
7.2.2 Particle Swarm Optimization-Based Keyword Selection	128
7.3 Experiments and Results	131
7.3.1 Experimental Setup: Baseline Approach	132
7.3.2 Experimental Setup: Similarity Heuristic-Based Key- word Selection Approach	132
7.3.3 Experimental Setup: Particle Swarm Optimization- Based Keyword Selection Approach	133
7.4 Results	134
7.5 Discussion	136
 8 Conclusions of the Thesis	 138
8.1 Summary of Thesis	138
8.2 Research Conclusions	141
8.3 Research Novelty and Contributions	143
8.4 Limitations and Directions for Future Work	144
 Appendices	 173

List of Figures

1.4.1 Twitter Syndromic Surveillance Pipeline	7
2.2.1 Example parse tree for the sentence <i>Fed raises interest rates</i>	17
2.3.1 Illustration of Label Propagation	26
2.4.1 Illustration of RNN architecture [71]	28
2.4.2 Illustration of CNN architecture [227]	31
2.5.1 Example 3-dimensional vector space [228]	32
2.6.1 Illustration of word2vec architectures [177]	35
2.6.2 Illustration of the Distributed Memory Model of Paragraph Vectors (PV-DM) [147]	36
2.6.3 Illustration of the Distributed Bag of Words version of Para- graph Vector (PV-DBOW) [147]	37
3.2.1 PRISMA flow diagram for the identification and selection of studies	42
3.3.1 Word cloud of statistical and machine learning methods dis- covered in review	42
3.3.2 Breakdown of studies by country	43
3.3.3 Most studied diseases each year. ¹	44
3.3.4 Most applied algorithms each year	45
3.3.5 Bubble chart showing the trends of research activity in public health application domains with time.	46
3.3.6 Most applied algorithms each year	47
4.2.1 Map of a Tweet as obtained from the Twitter API	64
5.4.1 Attention-based RNN model	86
6.2.1 Bar chart showing emoji frequency in labelled data	97
6.3.1 Graph of F2 performance of Iterative Labelling using differ- ent confidence thresholds	104

6.3.2 Graph showing how many correct assimilations the iterative labeling algorithms make per iteration using labelled data from a different time period	105
6.4.1 Time taken to perform relevance classification on a collection of Tweets.	107
6.5.1 Plot of Tweets representative of distances in attention embedding space. The axes represent t-SNE dimensional values.	111
6.5.2 Heatmap showing weights placed on words in a Tweet by our attentive bi-RNN model	111
6.6.1 Comparison for Twitter signal extraction using LSTM relevance filtering	113
6.6.2 Comparison for Twitter signal extraction using ABLSTM relevance filtering	113
6.6.3 Comparison for Twitter signal extraction using Self Training relevance filtering	114
6.6.4 Comparison for Twitter signal extraction using Co-Training relevance filtering	114
6.6.5 Comparison for Twitter signal extraction using MLP relevance filtering	115
6.6.6 Comparison for Twitter signal extraction using LSTM relevance filtering	115
6.6.7 Comparison for Twitter signal extraction using ABLSTM relevance filtering	116
6.6.8 Comparison for Twitter signal extraction using Self Training relevance filtering	116
6.6.9 Comparison for Twitter signal extraction using Co-Training relevance filtering	117
6.6.10 Comparison for Twitter signal extraction using MLP relevance filtering	117
6.6.11 Comparison of Deep Learning Twitter signal extractions with control signal	118
6.6.12 Comparison of Iterative Labelling Twitter signal extractions with control signal	118
6.6.14 Chart of mean UK temperature for the time period of summer 2016 ²	121
7.2.1 Flow chart for simple trial and error keyword selection	125
7.2.2 Illustration of different problem search spaces.	128
7.2.3 Illustration of PSO particles in a search space	129
8.1.1 Time series plot showing generated Twitter syndromic surveillance signal with a ground truth signal for comparison	139

List of Tables

3.1	Inclusion and exclusion criteria.	41
3.2	Summary of statistical and machine learning methods and data sources for surveillance using Twitter data	48
3.3	Summary of statistical and machine learning methods and data sources for event detection using Twitter data	51
3.4	Summary of statistical and machine learning methods and data sources for pharmacovigilance using Twitter data	54
3.5	Summary of statistical and machine learning methods and data sources for forecasting using Twitter data	56
3.6	Summary of statistical and machine learning methods and data sources for disease tracking using Twitter data	58
3.7	Summary of statistical and machine learning methods and data sources for geographic identification using Twitter data	59
4.1	Information on the data corpus collected before cleaning	66
4.2	Availability of geolocation attribute in collected Twitter Data	66
4.3	Information on the data corpus collected after cleaning	68
4.4	Our list of word classes with their member words	71
4.5	Distribution of constructed features and classes across the dataset	74
4.6	A random sample of words and their 8 most similar words as computed from a Twitter dataset using Skipgram embeddings.	75
4.7	A random sample of Tweets and their 2 most similar Tweets as computed using PV-DM embeddings.	76
6.1	F1 scores after feature ablation	96
6.2	Most informative words measured by their <i>Informativeness</i> and their relevant:irrelevant prior probabilities	98
6.3	Most frequent emojis in labeled data and their distributions	99
6.4	Classification performance of different Tweet feature representations obtained from deep embeddings	100

6.5	Results of relevance classification on the test data. Naive Bayes (NB), Decision Tree (DT), Random Forest (RF) Logistic Regression (LR), Support Vector Machine (SVM) and Multilayer Perceptron (MLP) algorithms are reported together with the self-training and co-training iterative labelling algorithms.	101
6.6	Confusion matrix for MLP fully-supervised classification on the test data	102
6.7	Confusion matrix for Co-training iterative labelling algorithm on the test data	102
6.8	Performance of Generative Classification Network with baselines on relevance filtering task.	107
6.9	Performance of Attentive Bi-directional Recurrent Neural Network and baselines on Tweet relevance classification task. . .	110
6.10	Pearson correlations and P-Values for detected signals with syndromic surveillance signals	119
7.1	Keywords at each iteration of the Similarity Heuristic Keyword Selection Process	133
7.2	Performances of different approaches to keyword selection . .	135

List of Algorithms

1	Iterative labelling Algorithm	80
2	Heuristic-Based Automatic Keyword Selection	126

Acknowledgements

Chapter 1

Introduction

1.1 Twitter Mining for Syndromic Surveillance

We are currently living in the golden age of information. The internet and big data has become prominent in many of our lives. Today, people are interconnected and have a web (or social media) presence which reflects a bit of their lives, experiences and who they are. Public Health England (PHE) is always looking to improve on the current monitoring system and is particularly keen to investigate the use of additional data sources to increase the sensitivity and specificity of alarms. Web activity and social media data may capture aspects of behaviour that are not captured by more traditional data sources. In fact, the use of data from social media sites such as Facebook or Twitter has been gaining momentum for disease surveillance. Furthermore, in developing countries where access to medical experts may be restricted but where use of mobile phones and social media is becoming more common, it is possible that such data may provide insights into the health of the population that are not otherwise available, alert to outbreaks and also provide a platform to spread information to combat such outbreaks.

Twitter and social media as a means of public health monitoring is not intended to replace traditional forms of syndromic surveillance, but rather augment it. Twitter data could be useful as an additional data source, offering us insight into a different demographic of the general population, who have different health reporting behaviours. If required, such as in scenarios where traditional syndromic surveillance is reduced or limited in operation,

like periods of public strikes or countries with reduced infrastructure, Twitter data could also be used as a proxy for traditional syndromic surveillance.

There already exists research into the utility of social media data for public health varying from global models of disease, to the prediction of an individual's health and when they may fall ill [226]. Ginsberg et al. [84] put forward an approach for estimating Flu trends using the relative frequency of certain Google search terms as an indicator for physician visits related to influenza-like symptoms. This was possible because the relative frequency of certain queries is highly correlated with the percentage of physician visits in which a patient presents influenza-like symptoms. They found that there was a correlation between the volume of these Google search terms and the recorded Influenza-Like Illnesses (ILI) physician visits reported by the Centre for Disease Control (CDC). Harrison et al [226] carried out an investigation into whether online Yelp reviews might signal threats to food safety. They reported that their results suggest that Yelp surveillance may identify small outbreaks of foodborne illness that traditional surveillance techniques miss.

De Quincey and Kostkova [67] introduced the potential of Twitter in detecting outbreaks. They posited that the amount of real-time information present on Twitter, either with regards to users reporting their own illness, the illness of others or reporting confirmed cases from the media, is both rich and highly accessible. We believe that Twitter could be a potent source of syndromic surveillance data. Firstly, unlike (Google) search queries, Twitter provides full text posts which provide domain experts with more descriptive information. Also, Twitter profiles and posts contain semi-structured metadata (such as age or location) allowing for a more detailed statistical analysis. And, despite the fact that Twitter appears targeted to a young demographic, it in fact has quite a diverse set of users. The majority of Twitter's nearly 10 million unique visitors in February 2009 were 35 years or older, and a nearly equal percentage of users are between ages 55 and 64 as are between 18 and 24 [55]. Chen et al. [42] managed to distinguish different biological phases of the flu from the content of tweets using a temporal topic model. Many of the published work on tracking flu or ILI is based on the USA where the volume of Tweets is greatest. While some work has also been done in the UK [146], it is few and far between, and has usually been mirroring the research in the US which is predominantly concerned with the flu of ILIs.

Given the differences in health-seeking and health-reporting behaviour between age groups in a population, Twitter's popularity with the younger age groups who do not seek medical advice through traditional routes may give some insight into possible gaps in the current syndromic surveillance systems. As such, it offers us the opportunity to capture and understand health-seeking behaviour within these subsets of the population that may

not have been previously captured. In addition, given the real-time nature of the Twitter stream, syndromic surveillance using Twitter could provide a faster means of observing and identifying incidents. Also, it could alert us to the type of language colloquially used to express concern about different syndromes, increasing understanding of how the population may discuss or report incidents when they may not be interacting directly with the health care system. Finally, from our research, we learn that reported trends on Twitter can predict the trends observed by traditional syndromic surveillance systems. This means that Twitter has some potential for detecting public health incidents before traditional surveillance systems.

1.2 Research Questions

This project, inspired and motivated by the ideas described in section 1.1, set out to investigate the utility of Twitter, as a social media data source, for syndromic surveillance. A lot of the existing research in this area has been carried out in the US and focused on ILIs as a syndrome. We look towards not only investigating Twitter for syndromic surveillance in the UK, where the volume of Tweets is lower than in the US, but also examining its utility for the syndrome of asthma/difficulty breathing. In doing so, we seek to answer the following questions with our research:

Is there useful, extratable information in the large amounts of Twitter data available? We know that enormous volumes of social media data is being produced everyday. We begin with the simple hypothesis that there is value in this big data, as the data is so expansive that statistically speaking, there must be something of use among it. As part of the initial stages of this project, we encountered literature, along with our own experiments, that confirmed this hypothesis. However, this leads on to our next question.

How can useful information effectively and efficiently be extracted from Twitter? This question has been at the center of this project, as a lot of the time and effort was dedicated towards developing techniques for mining textual Twitter data. Partly owing to this question, this project contains overlapping elements of various computing and scientific fields including data mining, machine learning, pattern recognition and information retrieval. As these are all well-developed fields of research, we have the advantage of having access to a plethora of established techniques. As such, we make use of state-of-the-art algorithms and techniques, but also try to achieve novel adaptations or improvements in methods or algorithms in relation to our work.

Does the information extracted from Twitter mirror the real-world such that it is a reasonable data source for syndromic

surveillance? While we can measure the ability of our proposed methods and algorithms to extract information from Twitter, success in such an evaluation does not equate to success in mirroring the public health state. This is because despite us evaluating our results against real-world syndromic surveillance data, we cannot be sure that such syndromic surveillance data is capturing any health care event entirely, or even that we are dealing with non-overlapping subsets of the population.

These questions informed the aims and objectives of this project which we present in the following section

1.3 Research Aims and Objectives

As was mentioned previously, research into Twitter data for syndromic surveillance has been predominantly focused around the study of ILIs as a syndrome in the US. This PhD project is based in the UK and looks at the syndrome of *asthma/difficulty breathing*. This is an interesting syndrome as it is a non infectious disease in contrast to ILIs, as well as the other diseases commonly studied in relevant literature, which focus on infectious diseases. This is understandable as the spread of dangerous infectious diseases can be very debilitating for a country. However, that is not to say that there is no need to study non-infectious diseases. An uptick in reports for non-infectious diseases is still a cause for concern as this could be a sign of biological attack, adverse meteorological phenomena or even a social or infrastructural failure, which all need to be detected and understood for the sake of public health and safety. Hence, our aim is **to establish if social media data, and specifically, Twitter data can be used in the context of syndromic surveillance in order to generate or contribute to alarms for a specific (non-infectious) syndrome - *asthma/difficulty breathing* - in the UK..** To this end, we carried out this project with the following objectives:

- i To conduct a comprehensive review to evaluate state-of-the art techniques that have the potential to improve our ability in detecting and understanding events.
- ii To conduct a practical study using a number of past events detected by existing syndromic surveillance systems to assess the value of alternative data sources.
- iii To investigate the challenges of capturing, storing and handling real-time data in the context of syndromic surveillance.
- iv To investigate novel data modelling techniques that can be used with Twitter data to generate public health alarms.

1.4 Research Contributions and Outputs

The research carried out as part of this PhD project makes the following contributions:

- A thorough study mapping the field of Twitter mining for public health and syndromic surveillance purposes. This study identified and described the various ways in which Twitter has been mined for health purposes, as well as algorithms, ideas and approaches implemented.
- The introduction of the use of semi-supervised classification for Tweet classification in the context of syndromic surveillance.
- Highlighting the efficacy and discriminatory power of emojis in text classification problems, together with capable features for Tweet classification within the context of syndromic surveillance
- A different and experimental approach to text classification based on deep generative neural network models.
- An attention-based bi-directional Recurrent Neural Network model for syndromic surveillance.
- A novel general framework for the automatic and optimal selection of keywords for Twitter data collection based on evolutionary algorithms and deep learning.

These contributions have led to the following peer-reviewed publications first-authored by Oduwa Edo-Osagie during this PhD project:

- Edo-Osagie, Oduwa, et al. "Twitter mining using semi-supervised classification for relevance filtering in syndromic surveillance." *PloS one* 14.7 (2019): e0210689.
- Edo-Osagie, Oduwa, et al. "Deep learning for relevance filtering in syndromic surveillance: a case study in asthma/difficulty breathing." *International Conference on Pattern Recognition Applications and Methods. No. 8*. 2019.
- Edo-Osagie, Oduwa, et al. "Attention-Based Recurrent Neural Networks (RNNs) for Short Text Classification: An Application in Public Health Monitoring." *International Work-Conference on Artificial Neural Networks. Springer, Cham*, 2019.
- Edo-Osagie, Oduwa, et al. "A Scoping Review of the use of Twitter for Public Health Research". Currently under review for publication in the *European Journal of Public Health* journal.

Through our research efforts, we developed methods for a Twitter syndromic surveillance pipeline which we present in this thesis. This pipeline is shown in figure 1.4.1. It involves setting up a reliable stream from Twitter

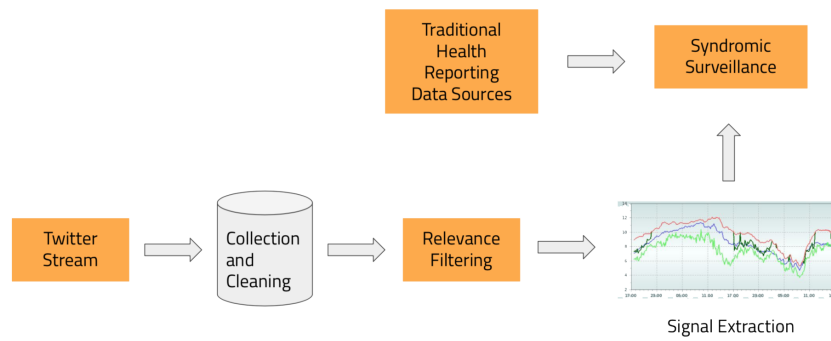


Figure 1.4.1: Twitter Syndromic Surveillance Pipeline

with which we can access data in real-time. This data is then subject to necessary cleaning and pre-processing operations. The cleaned data is then passed through our relevance filtering algorithms which allow us to extract information pertinent to distress over our syndromes of concern. Following this, a signal is extracted from the relevant data which can be potentially joined with data from other syndromic surveillance systems and used to help understand the public health state.

1.5 Big Data and Social Media

Mankind has seen rapid developments in data communication, storage and computation infrastructure in recent times. Due to the availability and proliferation of reliable and affordable computing hardware, a large percentage of daily human activity is connected to computers in some way. Each of these daily activities creates, and itself consists, data. It is estimated that 2.5 quintillion bytes of data are created each day from our collective activities [72]. This grand scale of data creation consequentially, requires the means for manipulation on a similar scale. These grand amounts of data and the ideas and techniques involving it are commonly referred to as “big data”.

The World Wide Web is arguably one of the most important and pervasive tools in modern everyday life. As much of our daily activities involve computers, these activities are usually also supported by the internet. The internet is changing the way we work, spend our leisure time and communicate with one another. It is estimated that the number of internet users worldwide reached 3.4 billion in 2016 [184]. An accompanying development that arguably, has been intertwined with the rise in internet use is that of mobile phones. Most mobile phones have internet capabilities. As such, they offer reliable, portable and readily available internet access at all times. Every large technological development in history has had an impact on the behaviour of society. Take the television as one simple example. Not only

did the television change the way family units spent time together, but it also created entire new industries. The onset of the internet is no different. In fact, as the internet has evolved, the manner in which people interact with it has evolved in kind. For instance, while it would have once been considered strange behaviour to eschew privacy, and broadcast the ins and outs of one's life, today, it is becoming a mainstream approach to life [174]. Social media is one of the forms in which this phenomenon has appeared in.

Social media refers to online user-generated content created and shared within a network or community of people. The creation of social networking websites such as MySpace (in 2003), and Facebook (in 2004) popularised the concept. Today, many years later, the social network is thriving harder than ever with many different platforms such as Instagram, Twitter, Snapchat, TikTok etc. each with their own offerings and idiosyncracies. One of the results of this is the fact that there now exist enormous streams of data created which has already attracted the attention of politicians [52], social scientists [190] and economists [25]. Social media creates public streams of communication, and scientists are starting to understand that such data can provide some level of access into the people's opinions and situations.

1.6 Syndromic Surveillance

Surveillance, described by the World Health Organisation (WHO) as “the cornerstone of public health security” [280], is aimed at the detection of elevated disease and death rates, implementation of control measures and reporting to the WHO of any event that may constitute a public health emergency or international concern. Disease surveillance systems often rely on laboratory reports. More recently some countries such as the UK and USA have implemented a novel approach called “syndromic surveillance”, which uses pre-diagnosis data and statistical algorithms to detect health events earlier than traditional surveillance [77]. Syndromic surveillance can be described as the real-time (or near real-time) collection, analysis, interpretation, and dissemination of health-related data, to enable the early identification of the impact (or absence of impact) of potential human or veterinary public health threats that require effective public health action [255]. For example, they use emergency department attendances or general practitioner (GP, family doctor) consultations to track specific syndromes such as influenza-like illnesses (ILI).

The current syndromic surveillance systems in the UK are advanced and have proved their worth, for example in the context of the Olympic Games that took place in London in 2012 [237]. They monitor syndromes such as seasonal flu, air pollution, flooding, heatwaves, etc. Reports are produced on a weekly basis, with alerts raised when statistical signals for each specific syndrome reach certain levels. Statistically significant aberrations or

“signals” are investigated to determine their public health importance. If deemed appropriate by human experts, alerts are converted to alarms and appropriate action taken.

As syndromic surveillance is concerned with the detection and understanding of public health threats, there is interest in rich, interesting and efficient data sources. As such, in addition to clinical data sources, there are sometimes investigations into alternative data sources. Examples of such alternative data sources include school and work absenteeism information, over-the-counter medication sales and animal illnesses or deaths [103]. Such alternative sources can be vital to improving existing syndromic surveillance solutions. They can offer advantages over traditional clinical data. For example, looking towards electronic or internet based data could provide a passive yet flexible system. While manual and traditional data sources can be detailed and rich, they can be labour-intensive and require human intervention and frequent maintenance. In this thesis, we turn our attention towards internet and social media data as a potential data source for syndromic surveillance.

1.7 Organisation of Thesis

The remainder of this thesis is organised as follows: In chapter 2, we provide a background, through which the reader is familiarised with the concepts with which this thesis is involved. We build on this in chapter 3, where we present a comprehensive scoping review of the use of Twitter for public health research. In that chapter, we make use of the PRISMA framework to map the literature and understand the current state of affairs. With this understanding, we then identify gaps in the literature which inform the thesis in its following chapters. In the first part of chapter 4, we describe the Twitter platform and Application Programmer Interface (API) with which we interact with the platform. We then describe our process of collecting, processing and storing Twitter data in the second part. We also discuss a number of feature extraction and representation techniques, (some of which are novel contributions of our work), which we apply to the collected Twitter data. In chapter 5, we discuss the issues necessitating further relevance filtering of collected Twitter data, in order to build a signal for syndromic surveillance. We then propose and describe semi-supervised learning techniques to accomplish this. In chapter 6, we carry out and describe synoptic empirical studies with the goal of evaluating our proposed approaches to Tweet classification and relevance filtering for syndromic surveillance, and discuss the observed results. The final contribution is presented in chapter 7. There, we propose two frameworks facilitating the automatic and optimal selection of keywords to be used for capturing and collecting relevant Twitter data. We empirically evaluate these frameworks and compare them to standard manual human keyword selection. Finally, the thesis is

concluded in chapter 8, where our results and contributions are discussed, referring also to possible future work.

Chapter 2

Technical Background

We view the research question posed in this thesis broadly as a data mining application problem. However, due to the descriptive and narrative nature of the social media data in which we are interested, we will also be dealing with large amounts of user-generated textual data. For this reason, it may also be useful to view the problem as related to the field of natural language processing (NLP). In this chapter, we first investigate textual data mining applications and paradigms. After this, we take a look at techniques and activities carried out to process and understand text data by delving into the field of natural language processing. We then explore the field of machine learning which can be, and is applied to text data. Finally, we explore deep learning, a relatively new field of machine learning which has raised expectations in data science and artificial intelligence due to its success in lots of machine learning tasks.

2.1 Text Mining

Text mining, also known as text data mining or knowledge discovery from textual databases[102] generally refers to the process of extracting interesting patterns or knowledge from unstructured text documents and is an extension of data mining. It usually involves first structuring the input text using a variety of techniques and potentially adding or removing some linguistic features deemed important or unimportant. After this, patterns are derived within the newly structured data which can then be evaluated and interpreted to give some output. Below are some of the main examples of text mining tasks

2.1.1 Text Categorization

Text categorization (also known as *document classification* or *text classification*) is a task where the aim is to assign a document to one or more predefined categories or classes. The automated categorization (or classification) of texts into predefined categories has seen an increased interest in the last ten years, due to the rapidly growing availability of documents in digital form [240] and the ensuing need to organize them. Until the late '80s the most popular approach to text categorization was a knowledge engineering one, involving manually defining a set of rules encoding expert knowledge on how to classify documents under the given categories [213, 48, 121]. From the '90s, such rule based classifiers lost popularity in favour of machine learning based approaches. The advantages of these approaches are an accuracy comparable to that achieved by human experts, and a considerable savings in terms of expert manpower, since no intervention from either knowledge engineers or domain experts is needed for the construction of the rules for the classifier or for its porting to a different set of categories or classes [233]. Today, text categorization in its essence, is the classic statistics and machine learning classification problem specialized for textual data. The idea is to be able to do this automatically, that is, without any human intervention. To accomplish this, a dataset of text documents is collected. None or some of the documents in this dataset could already be categorized and have a label associated with them. A statistical model is then built on the collected data which can be seen as approximating a function for determining a document's class label. The statistical technique used to construct this model is what determines whether the dataset needs to be labeled or not. There are many applications of text classification in the commercial world [112] with email spam filtering now being the most ubiquitous. Other applications of text categorization are genre classification and readability assessment[208, 51].

2.1.2 Sentiment Analysis

Sentiment analysis (also known as *opinion mining*) refers to the systematic process of identifying and extracting subjective information from textual data. In its simplest form - classifying the polarity (positive, negative or neutral) of a given text excerpt or document - sentiment analysis is an offshoot of text categorization/document classification. However, sentiment analysis is not necessarily always as simple as that. The document's sentiment may be classified against a multi-point scale. Pang and Lee[198] expanded on the task of classifying a movie review as either positive or negative to predict star ratings on either a 3 or a 4-star scale. Snyder and Barzilay[246] performed an analysis of restaurant reviews in order to predict ratings for various aspects of a given restaurant, such as the food and atmosphere on a 5-star scale. While it has been a popular area of research and commercial implementation recently, there has been a steady interest

in sentiment analysis since the 1980s [275]. Early work in this area focused mostly on interpretation of metaphor, narrative, point of view, affect and evidentiality in text [101, 110, 128] and could be viewed as a forerunner of the field as it is today. From around 2001 however, there occurred a widespread awareness of the research problems and opportunities that sentiment analysis raises [35, 62, 63, 181, 188, 274, 284]. Since then, there have been hundreds of papers published on the subject.

Also, while it is similar to text categorization, sentiment analysis poses a new set of challenges. In contrast with text categorization, in sentiment analysis, we often have relatively few classes that generalize across many domains. In addition, while the different classes in topic-based text categorization can be completely unrelated, the target classes in sentiment analysis typically represent opposing categories, if the task is structured as a binary classification, or different intensities of one objective on a sliding scale, in the case of multi-class classification [199].

In tackling syndromic surveillance through Twitter data, we perform relevance filtering of Tweets. While this task is similar to sentiment analysis as it is also a classification task, the techniques used are not always directly transferable. While sentiment analysis aims to establish the emotional polarity of a text, relevance filtering aims to achieve semantic understanding of the text. Establishing whether the tone of a Tweet is positive or negative is secondary to the task of relevance filtering, which needs to understand the intention of the Tweet. Knowing whether a Tweet is positive or negative alone does not help establish its relevance for syndromic surveillance. Secondly, while some of the techniques we propose for solving the problems of relevance filtering and syndromic surveillance work well at such tasks, they would not lend themselves well to the sentiment analysis task. One example of such an algorithm is the Generative Classification Network, which is discussed in section 5. However, the novel automatic keyword search algorithm we propose in section 7 is not only relevant to our work, but could also be useful in a sentiment analysis scenario.

2.1.3 Information Retrieval

Information retrieval can be defined as the process of finding unstructured data (usually of a textual nature) that satisfies an information need, within large collections of data. The term *information overload* (or *infobesity*) which refers to the difficulty of understanding an issue and effectively making decisions when one has too much information about that issue [283] is the reason information retrieval is so important today. With the vast amounts of ever-growing data in the world today, information retrieval tries to provide efficient representation, storage, organization of, and access to information items. Information retrieval can be a bit tricky because it usually deals with natural language text which is not always well structured and could be semantically ambiguous. Information retrieval systems can be distinguished

by their scale of operation. An example of large scale information retrieval is a web search engine. Here, the information retrieval system has to provide search over billions of documents. An important issue for systems operating at this scale is efficiently storing and indexing documents. In addition to this, such systems may need to efficiently distribute files across devices if distributed computing is used. Conversely, an example of small scale information retrieval is *personal information retrieval*. Consumer applications and operating systems fall under this category and they increasingly provide search capabilities albeit ranging in sophistication. Some examples of these are search in email client application and Spotlight Search on Mac OS X[10]. Issues here include handling the broad range of document types on a typical personal computer, and making the search system maintenance free and sufficiently lightweight in terms of startup, processing, and disk space usage that it can run on one machine while still providing a good user experience. In between these two categories are the middle scale information retrieval systems. Such systems involve institutional or domain-specific search engines. In order to effectively retrieve relevant documents, the documents are usually transformed into suitable representations. There are a number of different information retrieval strategies and each of these strategies incorporates a specific model for its document representation purposes. The models differentiating these strategies are as follows:

- **Set-Theoretic Models:** These models represent the data as sets of words or phrases. This allows us to apply operations based in set theory on the data in order to compute values such as similarity. The main example of this kind of model is the standard boolean model of information retrieval [169]. This model is based on Boolean logic and classical set theory such that the documents to be searched and the user's query are conceived as sets of terms. Retrieval is based on whether or not the documents contain the query terms.
- **Algebraic Models:** Algebraic models represent documents as vectors or matrices. With this, linear algebra operations can be carried out on the documents and the similarity of the query vector and document vector is represented as a scalar value. An example of such a model is the vector space model which is explored in detail below.
- **Probabilistic Models:** These models view the process of information retrieval as a probabilistic inference problem. Similarities are computed as probabilities that a document is relevant for a given query.
- **Feature-based Models:** These are a sort of free-form model. Such models view documents as vectors of values or feature functions (or just features) computed on the document, and seek the best way to combine these features into a single relevance score. Such features can incorporate expert domain knowledge to formulate domain-specific features to look out for. Feature functions are arbitrary functions of

document and query, and as such can contain any of the above models as part of it.

2.1.4 Text Clustering

Text Clustering (or document clustering) refers to the process of automatically grouping bodies of textual data, usually based on their content similarity, into previously undefined categories or classes. The problem of text clustering can be defined as follows: Given a set of n documents noted as D and a predefined cluster number K , D is split into K document clusters $\{D_1, D_2, \dots, D_K\}$ with $D_1 \cup D_2 \cup \dots \cup D_K = D$ so that the documents in the same document cluster are similar to one another while documents from different clusters are dissimilar. Text clustering was initially developed to improve the performance of search engines through pre-clustering the entire corpus [58]. Text clustering later has also been investigated as a post-retrieval document browsing technique [54, 58, 153]. Research into the clustering problem precedes its applicability to the text domain. Traditional methods for clustering have generally focused on the case of quantitative data [89, 113, 130] and categorical data [8, 82, 90]. A broad overview of clustering is given later below. Text clustering however, poses a slightly different challenge in that the dimensionality of the data is very high (because each document consists of many terms) and sparse (because the terms present in a document will only consist of a relatively small sample of the total vocabulary of terms). The standard clustering algorithms can be categorized into *partitioning algorithms* such as k-means or k-medoid and *hierarchical algorithms* such as Single-Link or Average-Link [130]. **Scatter/Gather** [58] is a well-known hybrid algorithm which has been proposed for text clustering that uses both approaches to clustering. It uses a hierarchical clustering algorithm to compute an initial clustering which it then refines using the k-means clustering algorithm. However, the above methods of text clustering algorithms do not really address the special challenges of text clustering [15]. This has motivated the development of clustering methods tailored to text data such as SuffixTree Clustering [291] and frequent term-based clustering [15]. The clustering problem finds numerous applications in customer segmentation, classification, collaborative filtering, visualization, document organization, indexing and discovering meaningful implicit subjects across a body of documents.

2.2 Text Processing: Natural Language Processing

Natural Language Processing (often abbreviated to NLP) is a field in computer science that is concerned with the automatic (or semi automatic) processing of natural human language by a computer rather than in a specialized artificial computer language. Natural Language Processing is a very broad field consisting of signal and speech processing and recognition, syntactic analysis, semantic analysis and pragmatics to name a few. However, for our purposes, we are really only concerned with the aspects of natural language most applicable to processing text data. With this in consideration, we take a look at *syntactic analysis* and *semantic analysis*. Before going into this any further, we would like to define some linguistic terminology:

- Morphology: This refers to the structure of a word. For example, *undoubtedly* can be thought of as composed of a prefix *un-*, a stem *doubted*, and an affix *-ly*.
- Syntax: This refers to the way words are used to form phrases and sentences and is heavily concerned with language grammar and its rules. e.g., it is part of English syntax that a determiner such as *the* will come before a noun.
- Semantics: Semantics can be distinguished into two types: *compositional semantics* and *lexical semantics*. Compositional semantics refers to the derivation of meaning based on syntax. Lexical semantics on the other hand, is concerned with the meaning of individual words.
- Pragmatics: This is concerned with the meaning of words and phrases in different contexts.

Now we can take a closer look at the tasks with which natural language processing is concerned:

2.2.1 Syntactical Analysis

Parsing

Parsing is a process of analyzing a sentence (or string of words) by taking each word and determining its structure from its constituent parts. Parsing process makes use of two components: a *parser* and a *grammar*. Before parsing natural language data, a grammar must first be established. A grammar is a set of structural rules governing the composition of clauses,

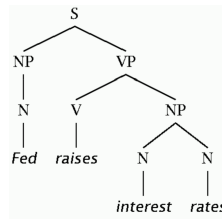


Figure 2.2.1: Example parse tree for the sentence *Fed raises interest rates*

phrases, and words in any given natural language. The parser is a procedural component and is a computer program that can build a constituency graph for a sentence using a specified grammar. The constituency graph shows the underlying phrase structure in a sentence. For example, employing a parser on the sentence *Fed raises interest rates Grammar* yields the parse tree (S (NP (N Fed)) (VP (V raises) (NP (N interest) (N rates)))) using bracket graph notation [261]. A visual representation of the tree is shown in fig 2.2.1.

Lemmatisation

Lemmatisation is the task of grouping together word forms that belong to the same inflectional morphological paradigm and assigning to each paradigm its corresponding canonical form called lemma. For example, English word forms *go*, *goes*, *going*, *went*, *gone* constitute a single morphological paradigm which is assigned the lemma *go* [81]. Simply put, it is the process of removing inflectional endings from a word in order to return it to its dictionary form. Lemmatisation is one of the techniques used in text mining to make sure that variants of words are not left out when texts are considered. It is a valuable preprocessing step in text mining. Traditionally, lemmatisation rules were hand-crafted. However, machine learning approaches to morphological analysis and lemmatisation became an increasingly interesting research subject. Jursic et al [125] treat lemmatisation as a classification problem. They treat lemmas as classes and lemmatise previously unseen word forms by classifying them as one of the lemma classes using a classifier previously trained on known word forms and their lemmas. Gesmundo and Samardzic[81] approach lemmatisation as a tagging problem and assigns to each word a label which encodes the transformation required to obtain the lemma string from the given word string. Chrupala[47] proposed a system which learned the lemmatisation rules from a corpus. The mappings between word forms and lemmas were encoded by means of the *shortest edit script*[185]. The sets of edit instructions are considered as class labels.

Stemming

Stemming is another one of the techniques used in text mining to make sure that variants of words are not left out when texts are considered. Stemming, similar to lemmatisation, is a technique used to reduce different grammatical forms of a word to its root form. Both of these processes reduce a word to a base form - *stem* in stemming and *lemma* in lemmatisation. The difference between the two processes is that the ‘stem’ is obtaining after applying a set of rules but without bothering about the part of speech (POS) or the context of the word occurrence. Studies based on stemming and lemmatization techniques have reported improved performance in a number of text mining tasks and are almost required as part of the preprocessing stage in tasks such as text clustering and text categorization [14, 119]. Since the lemmatization problem was first introduced in 1968 [162], there has been much research into stemming algorithms and as such, many different approaches have been proposed. Broadly speaking, these algorithms can be classified into three groups - **truncating**, **statistical** and **mixed**. Examples of truncating stemmers are the *Lovins stemmer*, *Porters stemmer*, *Paice/Husk stemmer* and the *Dawson stemmer* [163, 214, 215, 46, 64]. Examples of statistical stemming stemmers are the *Hidden Markov Model (HMM) stemmer* and the *Yet Another Suffix Stripper (YASS) stemmer* [175, 168]. Examples of mixed stemmers are the *Krovetz stemmer (KSTEM)* and the *corpus-based stemmer* [142, 282]. As of now, Porters stemming algorithm is the most widely used [5].

Sentence Boundary Disambiguation

Sentence boundary disambiguation (also known as sentence segmentation) refers to the task of identifying where sentences in a body of text begin and end. It is a surprisingly complex task as punctuation marks are often not a reliable marker of sentence boundaries. About 47% of the periods in the Wall Street Journal corpus denote abbreviations [249]. Approaches to sentence segmentation can be broadly divided into two classes: traditional rule-based approaches and machine-learning approaches. Traditional rule based approaches involve hand-crafting a set of rules for identifying a sentence boundary. Cutting et al[57] proposed such an approach. They tried to find sentence delimiters by tokenizing the text stream and applying a regular expression grammar with some amount of look-ahead, with a hand-crafted list of possible abbreviations and a list of exception rules. The machine learning-based approaches automatically learn a set of rules from a set of documents where the sentence boundaries have been pre-labeled. Riley[221] presents an early application of machine learning to SBD, investigating the use of decision tree classifiers in determining whether instances of full stops (periods, in American English) mark sentence boundaries [219].

Another example is the SATZ architecture which used a neural network to disambiguate sentence boundaries and achieves 98.5% accuracy [197].

2.2.2 Semantic Analysis

Part-of-Speech Tagging

Commonly referred to as POS tagging, part-of-speech tagging is the process of marking the words in a piece of text as being their corresponding part-of-speech (i.e. noun, verb, pronoun, preposition, adjective, determinant etc.) based on their meaning and context. Each tag corresponds to a part-of-speech and collectively form a *tagset*. More fine-grained tagsets than the one described above could exist. For example, the Penn Treebank uses a tagset of 36 tags¹. POS tagging is difficult because sentences can be ambiguous and it is often hard to determine the correct context. Take the string *Flies like a flower*. *Flies* could be referring to the insect being fond of flowers or the string could be a clause characterizing some subject as flying in a certain manner. Essentially, *flies* in that string could be a noun or a verb. Similarly, *like* could be a preposition or a verb in that string. There are three main approaches to POS tagging: rule-based POS tagging, transformation-based POS tagging and Probabilistic POS tagging. An example of a rule-based POS tagging approach is the ENGTWOL tagger [129]. An example of a transformation-based approach is the Brill tagger [29]. And an example of a probabilistic approach is *Trigrams'n'Tags (TNT)* [27]. POS tagging is used in text-to-speech programs to determine what syllables in a word need to be stressed when they are spoken. For instance, the word *object* has emphasis as **object** when used as a noun and **object** when used as a verb.

Named Entity Recognition

The term “Named Entity” was coined for the sixth Message Understanding Conference (MUC-6) [88]. At that time, MUC was focused on Information Extraction tasks in which structured information of company activities were extracted from unstructured text, such as newspaper articles. In defining the task, people noticed that it is essential to recognize information units like names, including person, organization and location names. Identifying references to these entities in text was recognized as an important task and was called Named Entity Recognition (NER). Over the years that passed,

¹https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Named Entity Recognition received a lot of attention from the natural language research community. Early Named Entity Recognition systems used hand-crafted rules to identify named entities [218]. Modern systems, however, use machine learning techniques to automatically learn these rules. The current dominant technique for named entity recognition uses supervised learning, with models such as Hidden Markov Models (HMM), decision trees, Maximum Entropy models (ME) and support vector machines [20, 235, 26, 12]. The idea of the supervised machine learning class of named entity recognition approaches is to study the features of positive and negative examples of named entities over a large collection of labeled documents and design rules that capture instances of a given type.

2.3 Machine Learning for Text Mining

Even though machine learning has been around for a long time, until the late 1980s, the common approaches to most text mining tasks were based on knowledge engineering or manually crafted rule-based systems. From the 1990s, that approach increasingly lost popularity in favour of the machine learning paradigm. In 1959, Arthur Samuel defined machine learning as “a field of study that gives computers the ability to learn without being explicitly programmed” [229]. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Machine learning approaches can be divided into two categories: *supervised learning* and *unsupervised learning*.

2.3.1 Supervised Learning

In supervised learning, data usually consists of examples (records of given attribute values) which are labeled by the class to which they belong. The task here, is to find a model - known as a classifier - that will enable a newly encountered instance to be identified as one of the classes to which the data might belong to. In text mining, this basically means taking a collection of documents that have been labeled for some feature, like categories or parts-of-speech. These documents are then used as a “training set” to produce a statistical model which can then be applied to new text. Below are some popular models used for such purposes.

Support Vector Machines

The Support Vector Machine is a machine learning tool that is usually used for classification tasks but can be extended to regression tasks as well. An SVM model is a representation of the training examples as points in space, mapped so that the examples of the separate classes are separated by a hyperplane (or set of hyperplanes in high dimension problems) while maximizing the distance from the hyperplanes to the nearest training point. Given a set D of n training points,

$$D = \{(x_1, x_2) \dots (x_n, y_n)\} \quad (2.3.1)$$

with a hyperplane described by the equation

$$\langle w, x \rangle + b = 0 \quad (2.3.2)$$

To minimize the margin of the hyperplane, the idea is to solve the optimization problem

$$\text{minimize } ||w|| \text{ such that } y_i(\langle w, x \rangle - b) \geq 1, i = 1 \dots n \quad (2.3.3)$$

For non-linearly separable scenarios, a loss function may be applied to the SVM. Usually the hinge-loss function is used for this. SVMs can be extended to regression problems by the introduction of an alternative loss function [244]. The alternative loss function must be modified to include a distance measure. Some possible loss functions are the quadratic function, Laplacian function and Huber function.

2.3.2 Unsupervised Learning

In unsupervised learning, the data is not labeled and the task is to uncover some hidden structure from the unlabeled data. Basically, unsupervised learning involves employing statistical techniques to tease meaning out of a collection of text without any pre-training. Below are some popular examples of unsupervised learning.

Clustering

Clustering (or cluster analysis) is a term that refers to methods for grouping unlabeled data. It is the unsupervised classification of observations or documents into groups (termed clusters). Documents in a valid cluster are more similar to each other than they are to documents that are members of a different cluster. While they are both somewhat similar, clustering is fundamentally different from classification. In classification (which is a supervised task), we are provided with a collection of pre-labeled (ie. pre-classified as belonging to one or more groups) observations or documents. The task is to use that information to automatically label new and previously unseen observations. In clustering, the task is to group a given collection of unlabeled patterns into meaningful clusters and in doing so, label unlabeled data. Clustering is a multi-disciplinary field of research with applications in biology, image processing and information retrieval [194, 114, 217]. Clustering algorithms can broadly be divided into two groups: *hierarchical clustering* and *partitional clustering*.

Hierarchical clustering is based on the idea that clusters can have sub-clusters and organizes data such that we can obtain a set of nested clusters that are organized as a tree. Each node (cluster) in the tree (except for the leaf nodes) is the union of its children (sub-clusters), and the root of the tree is the cluster containing all the objects [257]. Hierarchical clustering algorithms can be divided into two groups:

- **Agglomerative:** This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- **Divisive:** This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Most hierarchical clustering algorithms are based on the single-link [245] and complete-link [136] algorithms. These two algorithms differ in the way that they characterize the similarity between a pair of clusters. In the single-link method, the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from the two clusters. In the complete-link algorithm, the distance between two clusters is the maximum of all pairwise distances between patterns in the two clusters. In either case, two clusters are merged to form a larger cluster based on minimum distance criteria.

A *partitional clustering* is simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset. It obtains a single arrangement of the data instead of a structure. The main partitional clustering algorithm is the k -means algorithm. In k -means, We first choose k initial centroids, where k is a user-specified

parameter, representing the number of clusters desired. Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated based on the points assigned to the cluster. We repeat these assignment and update steps until no point changes cluster membership.

Hierarchical clustering is often portrayed as the better quality clustering approach, but is limited because of its quadratic time complexity ($O(n^2(\log n))$ for agglomerative clustering and $O(n^2)$ for divisive clustering [222]). In contrast, K-means and its variants have a time complexity that is linear in the number of documents ($O(n)$), but are thought to produce inferior clusters [250]. In order to circumvent these downsides, hierarchical and partitional clustering are sometimes combined like in the case of the *Scatter/Gather* algorithm described above in section 2.1.4.

Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (or *LDA* for short) is a generative probabilistic model usually applied to text data. It is a Bayesian model which means that like the Naive Bayes model described in section 2.3.1, it is based on Bayes' theorem. and was first introduced by Blei et al. [21].

It is a hierarchical model where each document in a text corpora is modeled as a mixture of topics. A topic is in turn modeled as a multinomial probability distribution over a set of words. The general idea of LDA is based on the hypothesis that when a person writes a document, they have certain topics in mind. Writing about a topic means picking a word with a certain probability from the set of words of that topic. A whole document can then be represented as a mixture of different topics. When the author of a document is one person, these topics reflect the author's view of a document and their particular vocabulary [139]. The LDA algorithm can be described as follows: Given a set of documents D , a set of topics Z and a vocabulary of words W , the goal of LDA is to predict which topic z , a given document d belongs to after considering its words w . It does this by estimating $P(w|d)$ in such a way that it is a function of z and d so that it tells us the probability that a topic z generated word w . $P(z|d)$ can be given by the product of $P(w|z)$ and $P(z|d)$. For simplicity, $P(w|z)$ can be thought of as the proportion of assignments to topic z over all documents that come from this word w . Similarly, $P(z|d)$ can be thought of as the proportion of words in document d that are currently assigned to topic Z . This gives

$$P(w_i|d) = \sum_{j=1}^Z P(w_i|z_i = j)P(z_i = j|d)$$

Note that the number of latent topics Z has to be defined in advance so that we can adjust the degree of specialization of the topics. LDA aims to estimate the topic–word distribution ($P(w|z)$) and the document–topic distribution ($P(z|d)$) from an unlabeled corpus of documents using Dirichlet priors for the distributions [139]. Usually *Gibbs sampling* [87] is used for this. It iterates multiple times over each word w_i in document d_i , and for each word, iterates through each topic, z , calculating the the topic–word distributions ($P(w_i|z_i, j)$) until convergence. Since its introduction, LDA has had many applications in fields such as information retrieval (particularly topic modeling)[33, 178, 191], image processing[79] and bioinformatics[212].

2.3.3 Semi-Supervised Learning

In semi-supervised learning, both labelled and unlabelled data are employed. Usually, in such scenarios, there is a small amount of labelled data and a large amount of unlabelled data. Semi-supervised approaches are motivated by the fact that unlabelled data is plentiful and cheap while labelled data can be expensive (in time, effort or sometimes monetary cost). The ultimate goal of semi-supervised learning is to build models which achieve better performance than one would observe using labelled or unlabelled data alone. In order for semi-supervised learning to be applicable, the data must usually meet some assumptions we make about its structure. Semi-supervised learning usually make one or more of the following assumptions [36]:

1. **Continuity assumption** Data points which are geometrically close to each other are more likely to share a label.
2. **Cluster assumption** Data for separable problems will form discrete clusters, and data points in the same cluster are more likely to share a label.
3. **Manifold assumption** The data points lie approximately on a manifold of much lower dimension than the input space.

Below are some popular approaches and ideas for semi-supervised learning.

Self-Training

Self-training is a heuristic method for semi-supervised learning and is the oldest approach, dating back to the 1960s [232]. Self-training starts with a set of labeled data, and builds a classifier in a fully supervised manner. The constructed classifier is then applied to the set of unlabelled data to yield labels. Classified instances with a classification confidence exceeding a certain threshold, are added to the labeled set. The classifier is then retrained on the new set of labeled examples, and the process is repeated until some satisfactory stopping condition is reached. Self-training makes an important assumption that the labelling classifier’s high-confidence predictions

are correct. There exist variations of the iterative labelling procedure and policy of self-training. For example, instead of incorporating only classified unlabelled instances with high classification confidence, in some cases, all classified unlabelled instances are assimilated. It is even possible to build on this further by not simply adding all classified unlabelled instances, but by also weighing the assimilated instances according to their classification confidences. There also exist some extensions to the self-training algorithm, such as *co-training*. Like self-training, co-training begins with a set of labelled and unlabelled data, and tries to increase the labelled set by assimilating instances from the unlabelled set. However, co-training makes use of two or more classifiers, each usually with a different view of the dataset, with each view conditionally independent from others [23]. Self-training approaches to semi-supervised learning are simple to use, but because they are wrapper algorithms, can also leverage the power of complex algorithms for the fully supervised classifier aspect, seeing the best of both worlds. However, because the labelling of previously unlabelled instances is automatic, early mistakes reinforce themselves throughout the self-training process.

Graph-Based Semi-Supervised Learning

Graph-based semi-supervised learning approaches assess and take advantage of similarity between labelled and unlabelled instances in order to grow the labelled set. To accomplish this, a graph is constructed from the set of labelled and unlabelled data. In this graph, nodes are specified by labelled and unlabelled instances, while edges are specified by the similarities between nodes. Graph-based approaches make the assumption that instances connected by a heavy edge should belong to the same class or label. More formally, the graph G can be represented as an ordered set of vertices, V , and edges, E .

$$G = (V, E)$$

V represents the set of vertices which includes both labelled instances, L , and unlabelled instances, U . E represents a set of edges which represent similarity between instances in the dataset. A distance metric such as euclidean distance could be used to measure similarity between instances. The label of each sample from L is propagated to its unlabelled neighbours.

Transductive Support Vector Machines

The support vector machine described above in section 2.3.1 can be extended to work in a semi-supervised context. Recall that SVMs attempt to learn the maximum separating hyperplane between the different classes of the labelled data. The transductive SVM (or TSVM), seeks the largest separation between labeled and unlabeled data [267]. TSVMs are a means of improving the generalization performance of SVMs using unlabelled data. The TSVM approach can be distilled into three main steps. First, we must

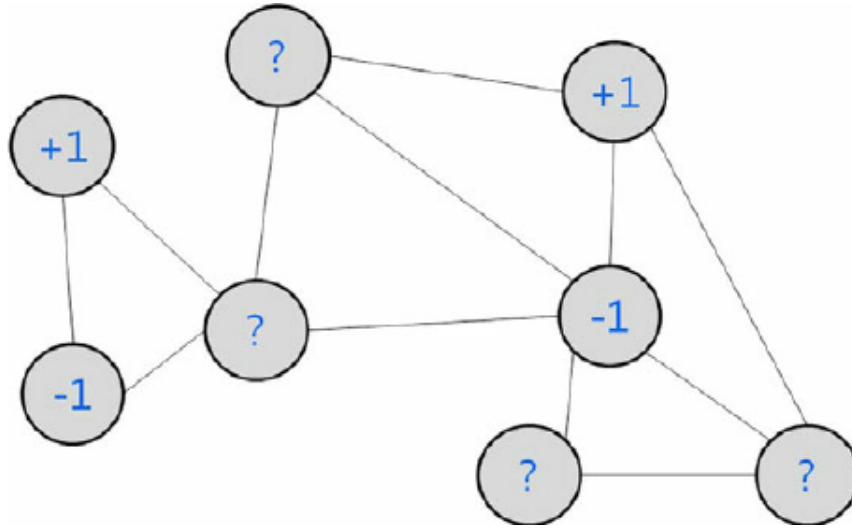


Figure 2.3.1: Illustration of Label Propagation

enumerate through all $|C|^{|U|}$ labellings for unlabelled instances, where C is the set of possible labels or classes and U is the set unlabelled instances. An SVM is then built for each $X_{\hat{u}}$ as well as each X_c , where $X_{\hat{u}}$ is the set of formerly unlabelled instances that had been assigned possible labels in the previous step, and X_c is the set of instances from the labelled set belonging to class c . Finally, the SVM with the largest separating margin for its hyperplane is selected. However, in application, it may be impractical to solve the TSVM for $|C|^{|U|}$ combinations when there is even a moderate amount of data involved. To address this issue, there exist heuristic such as deterministic annealing [223] and concave-convex programming [290]. Because, the TSVM focuses on a particular working set to achieve the optimal classification in that working set, it has been shown to sometimes have issues with generalization [75]. There exist some more variations to the SVM which attempt to solve the semi-supervised learning problem better than the standard TSVM including Progressive Transductive SVMs (or PTSVMs) [43], Laplacian SVMs [16] and SVMs with cluster kernels [37]. While transductive and other semi-supervised SVMs have the advantage of having a clear mathematical framework, optimization can be difficult. Additionally, they make more modest assumptions than self-training and graph-based semi-supervised approaches.

2.4 Neural Networks and Deep Learning

Artificial neural networks are a family of machine learning models inspired by the way that biological nervous systems, such as the brain, process information. A neuron is a cell that has several inputs that can be activated by some outside process. Depending on the amount of activation, the neuron produces its own activity and sends this along its outputs. In addition, spe-

cific input or output paths may be “strengthened” or weighted higher than other paths. The artificial neural network equivalent of a neuron is a **node**. A node receives a set of inputs, performs a weighted sum ϕ of these inputs, and passes the result to nodes further down the network. This operation can more formally be represented as such

$$\phi = \sum_{i=1}^n w_i \cdot a_i \quad (2.4.1)$$

where w_i is the weight for node i , a_i is the input to node i , and n is the number of nodes in a layer of the network. Many such layers are chained together to form a network, passing their outputs along. The network is trained with labeled data by feeding inputs into the network and fine-tuning the weights until the network always yields the expected class label as its output. This is not very different from regression in that parameters are being tuned to create a function that yields certain values. Hence, artificial neural networks can easily be adapted to solve regression problems.

Deep learning is a broad branch of machine learning concerned with learning representations for data through the application of a stack of consecutive non-linear transformations. The “deep” in the name “deep learning” refers to the number of layers through which the data is transformed. Modern deep learning is based on artificial neural networks. Shallow neural networks are usually distinguished from deep neural networks by having fewer than 2 layers.

Shallow architectures, particularly ones employing kernel-based learning suffer from the “*curse of dimensionality*”. Bengio and LeCun reviewed kernel-based learning and deep learning, producing a structured theoretical comparison [17]. They found that unless certain assumptions are met, methods relying on local kernels may need an exponential number of parameters to approximate the target function which relates the input to the output. Kernel based methods, such as the Support Vector Machine, are shown to have an expected error, which rises exponentially with the number of dimensions of the input space [97]. “*curse of dimensionality*” which such methods suffer from. However, It is worth noting that many practical problems adhere to the assumptions of kernel based models. Hence, these models are effective on these problems and may be the most suitable method to use in such cases.

Shallow learning usually relies on manual feature engineering. The input is subjected some preprocessing to simplify the input to represent the input data in a concise yet meaningful way, in order to circumvent the limitation highlighted above. The aim of feature engineering is to reduce the dimensions of the input data and create a (more) separable and smooth structure of the data such that a classifier can efficiently and effectively be applied to the data. However, this process can be domain-specific and relies on human intuition around the nature of the problem. Arguably, the biggest success

of deep learning is the fact that it is able to automatically learn representations of the raw data. In fact, deep learning is sometimes alternatively referred to as representation learning.

The relatively new field of deep learning has seen a lot of interest and activity. As a result, a lot of different architectures have been experimented with. Below, we discuss some of the most successful and well-researched architectures relevant to this thesis.

2.4.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a category of neural networks that incorporate sequential information. In such networks, connections between neurons, or nodes, form a directed graph along the input sequence. That is to say, while in a traditional neural network inputs are independent, in RNNs each node depends on the output of the previous node. This is particularly useful for sequential data such as text where each word depends on the previous one. While in theory, RNNs can make use of information in arbitrarily long lengths of text, in practice they are limited to looking back only a few steps due to the vanishing gradient problem. This problem is a phenomenon that occurs during the workings of the backpropagation algorithm, responsible for tuning the parameters of the network. Due to long sequences of matrix multiplications, gradient values shrink fast and gradient contributions from earlier neurons become zero. As a result of this, information from earlier inputs (words in the text) do not contribute to the overall algorithm. Long Short Term Memory (LSTM) [104] networks and Gated Recurrent Unit (GRU) [44] networks are flavours of the RNN architecture which make use of gating mechanisms to combat the vanishing gradient problem.

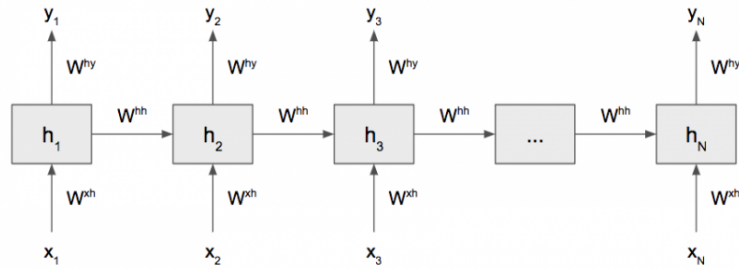


Figure 2.4.1: Illustration of RNN architecture [71]

Long Short Term Memory (LSTM) Networks

The LSTM model adds some complexity to the regular neural network architecture. The network has an input layer x , hidden layer h , LSTM cell state c and output layer y . Input to the network at timestep t is $x(t)$, output

is denoted as $y(t)$, hidden layer state is $h(t)$ and LSTM cell state is $c(t)$. The LSTM cell state is controlled by the gating mechanism as highlighted above briefly. Each cell consists of the following gates which interact with each other to dictate the overall cell state:

- input gate (i)
- forget gate (f)
- write gate (g)
- output gate (o)

Each of these gates has its own weights and biases and is a function of the previous timestep's hidden state $h(t-1)$. The hidden state of a layer can then be computed as a function of the cell state as shown below:

$$c(t) = f(t) \cdot c(t-1) + i(t) \cdot g(t) \quad (2.4.2)$$

$$h(t) = o(t) \cdot \tanh(c(t)) \quad (2.4.3)$$

For the sake of brevity and simplicity of our equations, let us assume that there is only one hidden layer l so that we do not have to specify different equations for the different edge cases that would come with multiple layers, such as when execution is in the first layer and has no previous layer or when it is in a middle layer or the final layer. In the real world scenario, this is not the case as each hidden layer state is influenced by the hidden state in the previous timestep as well as the state of the previous hidden layer. To adapt this, one may simply add the product of the weights and input of the previous layer to each activation function. The activation functions for the gates are computed as:

$$f(t) = \text{sigmoid}(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \quad (2.4.4)$$

$$g(t) = \tanh(W_{xg} \cdot x_t + W_{hg} \cdot h_{t-1} + b_g) \quad (2.4.5)$$

$$i(t) = \text{sigmoid}(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \quad (2.4.6)$$

$$o(t) = \text{sigmoid}(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o) \quad (2.4.7)$$

where W_{pq} are the weights that map p to q and b_p refers to the bias vector of p . For example, if we look at equation 2.4.4, W_{xf} refers to the weights going from input x to the forget gate f and so on while b_f refers to the bias of the forget gate f

Gated Recurrent Unit (GRU) Networks

The GRU is another solution for the short-term memory problem that simple RNNs possess, where they cannot properly update and learn weights for earlier inputs in a sequence. LSTMs and GRUs are very similar, the main difference is that GRUs have less parameters than LSTMs. Again, for the sake of brevity and simplicity of our equations, let us assume that there is only one hidden layer l . The GRU cell state is controlled by a gating mechanism, similar to the LSTM. Each cell consists of the following gates which interact with each other to dictate the overall cell state:

- update gate (z)
- reset gate (r)

The gates can be formalised as follows:

$$z(t) = \text{sigmoid}(W_{xz} \cdot x_t + W_z \cdot h_{t-1} + b_z) \quad (2.4.8)$$

$$r(t) = \text{sigmoid}(W_{xr} \cdot x_t + W_r \cdot h_{t-1} + b_r) \quad (2.4.9)$$

The hidden state of a layer is computed as a function of the input and gates as shown below:

$$h(t) = z(t) \cdot h(t-1) + (1 - z(t)) \cdot \tanh(W_x + r(t) \cdot W_h \cdot h(t-1)) \quad (2.4.10)$$

where W_{pq} are the weights that map p to q and b_p refers to the bias vector of p . For example, if we look at equation 2.4.8, W_{xz} refers to the weights going from input x to the update gate z and so on, while b_z refers to the bias of the update gate z and W_z refers to the weights for the update gate itself.

2.4.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a category of neural networks that have proven very effective for image classification [141]. CNNs introduce one or more convolutional layers, often with pooling layers for subsequent subsampling, which are then followed by one or more fully connected layers as in a standard multilayer neural network. This architecture is designed to consume 2D input which is why it is typically applied to images. The convolution layer involves applying one or more convolution filters/kernels to the input volume. The filter can be seen as a selector looking for a certain characteristic like a line, curve or shape. After this process, we are left with a flattened feature map of the input volume. The pooling layer is used to reduce the size of the feature map using a pooling filter. The pooling layer usually downsamples the convolved input by taking the

average or maximum value in an input region. Because of the consecutive pooling procedures in each layer, the output grows increasingly tolerant to small variations in the input. An invariance (or tolerance) to such changes is one of the core challenges in object recognition. While CNNs are typically applied to image data, they have also seen some success being applied to text data [134].

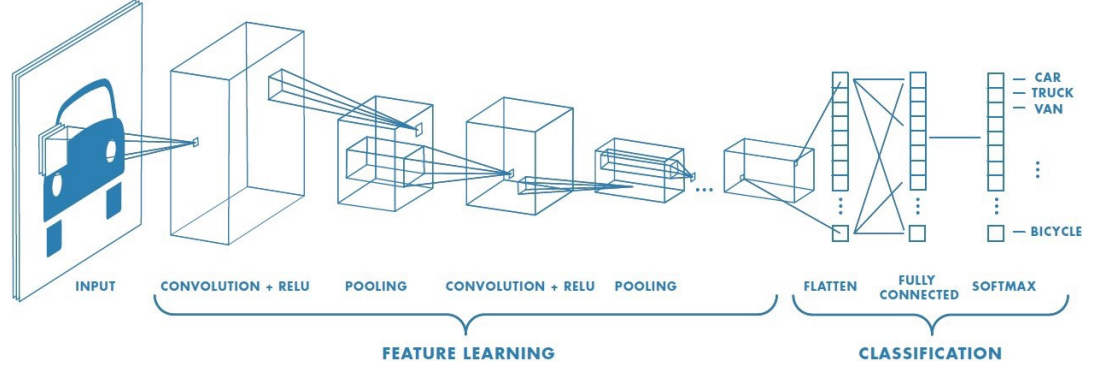


Figure 2.4.2: Illustration of CNN architecture [227]

2.5 Text Transformation and the Vector Space Model

As was mentioned in section 2.1, before text mining processes are carried out, the text data is usually first structured into a form better suited to complex computations. The most popular of such transformations is the *Vector Space Model*. The vector space model was developed for the SMART information retrieval system in the '70s [228]. The idea of the vector space model is to represent each document in a collection as a point in a space. Consider such a space D of n documents, with each document D_i represented by k (index) terms T with $\{k \in \mathbb{R} | k \geq 1\}$ forming a k -dimensional vector like

$$D_i = \{T_1, T_2, \dots, T_k\}$$

Figure 2.5.1 shows an example 3-dimensional space. Points that are near each other in this space are semantically similar and points that are far apart are semantically distant. An example similarity measure for two documents in such an arrangement would be to compute an inverse function of the angle between the corresponding vectors for the documents. If the documents are the same - their vectors are identical - the angle between them will be zero.

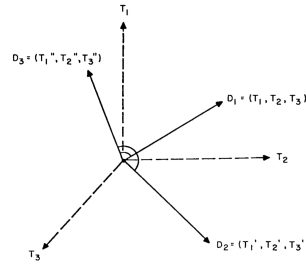


Figure 2.5.1: Example 3-dimensional vector space [228]

Not all uses of vectors count as employing a vector space model. Usually, the defining property of vector space models is that the values of their elements must be derived from event frequencies (eg. the number of times a word occurs in some, if any, context). Vectors are common in Artificial Intelligence and machine learning. In machine learning, a typical problem is to learn to classify or cluster a set of items. These items are usually represented as *feature vectors* [179, 278]. The novelty of the vector space model is that it uses frequencies of events in text corpora. Based on this, varieties of the vector space model have been organized into three groups: *term-document*, *word-context*, and *pair-pattern* [259].

2.5.1 The Term-Document Matrix

With a large collection of documents, (and hence, a large number of vectors), it is often convenient to organize the vectors into a matrix. In such a matrix, the column vectors represent the documents in the collection while the row vectors represent the terms (usually words) that could be in these documents. Such a matrix is called a *term-document matrix*. In the term-document matrix, each document vector represents the corresponding document as a bag of words. A bag is an unordered list without duplicates and the i -th element in the bag of words represents the count of the i -th word in some document. In information retrieval, the *bag of words hypothesis* is that we can estimate the relevance of documents to a query by representing the documents and the query as bags of words. That is, the frequencies of words in a document tend to indicate the relevance of the document to a query [259]. Such matrices are sparse because most documents will only use a small portion of the whole vocabulary leaving many zeros in the matrix.

2.5.2 The Word-Context Matrix

When the vector space model was introduced, Salton et al[228] focused on measuring document similarity, transforming queries into the vector space

in order to make them pseudo-documents. The relevance of a document to a query was given by the similarity of their vectors. Deerwester et al[69] observed that we could actually also measure word similarity, instead of document similarity, by looking at row vectors in the term–document matrix, instead of column vectors. This application of the vector space model is based on the *distributional hypothesis*. The distributional hypothesis in linguistics is that words that occur in similar contexts tend to have similar meanings [99]. In this space, a word may be represented by a vector whose elements are derived from the occurrences of the word in various contexts, such as windows of words and grammatical dependencies. Similar row vectors in the word–context matrix indicate similar word meanings.

2.5.3 The Pair-Pattern Matrix

In a pair–pattern matrix, row vectors correspond to pairs of words, such as mason:stone and carpenter:wood, and column vectors correspond to the patterns in which the pairs co- occur [259]. The pair-pattern matrix is based on the *extended distributional hypothesis* which states that patterns that co-occur with similar pairs tend to have similar meanings. Lin and Pantel[155] introduced the pair-pattern matrix, using it to measure the semantic similarity of patterns (ie. the similarity of column vectors) and reported that it could be used to determine when sentences are paraphrases of each other. Turney and Littman[258] proposed the use of the pair–pattern matrix for measuring the semantic similarity of word pairs (ie. the similarity of row vectors). They based it on the *latent relation hypothesis* which states that pairs of words that co-occur in similar patterns tend to have similar semantic relations [257] and is the inverse of the extended distributional hypothesis.

2.6 Learning Neural Representations of Text

Deep learning, applied with some of the techniques above has seen a lot of success in obtaining powerful text representations. Most popular of these are the word-context embeddings word2vec [177] and GloVe [205]. Both of these techniques are able to embed words in a vector space which manages to preserve semantic information. They make use of word contexts, building a statistical understanding of which words appear together and inferring the relationships between words in this way. However, these algorithms only work on words or tokens from a body of text. To obtain an embedding or vector representation for a whole sentence or document, one would have to perform some aggregative operation such as a sum or a mean of the vectors of constituent words. While this approach sometimes offers satisfactory performance, some information is lost in this process. To remedy this, there also exist some text embedding approaches operating on the sentence

or document embeddings. One such algorithm is paragraph2vec. Below, we describe these techniques in more detail.

2.6.1 Word2Vec

Word2Vec is an algorithm for embedding words in a meaningful vector space. It makes use of a two layer neural network trained to reconstruct the linguistic context of words. It is an unsupervised learning method which takes in large text corpora and builds this space with words from these corpora. There are two forms of word2vec, namely the Skipgram model and the Continuous Bag-of-Words (CBOW) model.

Skipgram

The Skipgram model is taught to learn word representations using a neural network with the training objective being to predict the surrounding words in a sentence or document. More formally, given a sequence of training words $w_1, w_2, w_3, \dots, w_T$, the objective of the skipgram model is to maximize the average log probability

$$\sum_{t=1}^T \sum_{-c \leq j \leq c} \log p(w_t + j | w_t) \quad (2.6.1)$$

where c is the size of the training context. Simply put, it is the window size, left and right of how many words to be considered.

Continuous Bag of Words (CBOW)

The continuous bag of words model is very similar to the skipgram model but has a slightly different training objective. While skipgram uses a given word to predict the other words around it, the continuous bag of words model uses a context of words to predict the word that they surround. Formally speaking, given a sequence of training words $w_1, w_2, w_3, \dots, w_T$, the objective of the continuous bag of words model is to maximize the average log probability

$$\sum_{t=1}^T \log p(w_t | \sum_{-c \leq j \leq c} w_j) \quad (2.6.2)$$

where c is the size of the training context.

The skipgram model and continuous bag of words model are illustrated in figure 2.6.1. By using a sum of words in a context to predict a target

word, CBOW smoothes over a lot of the distributional information (as a result of treating an entire context as one observation). For this reason, CBOW tends to work best with larger datasets and shorter sentences. The skipgram model treats each context-target pair as a new observation and is useful for smaller datasets, with longer sentences.

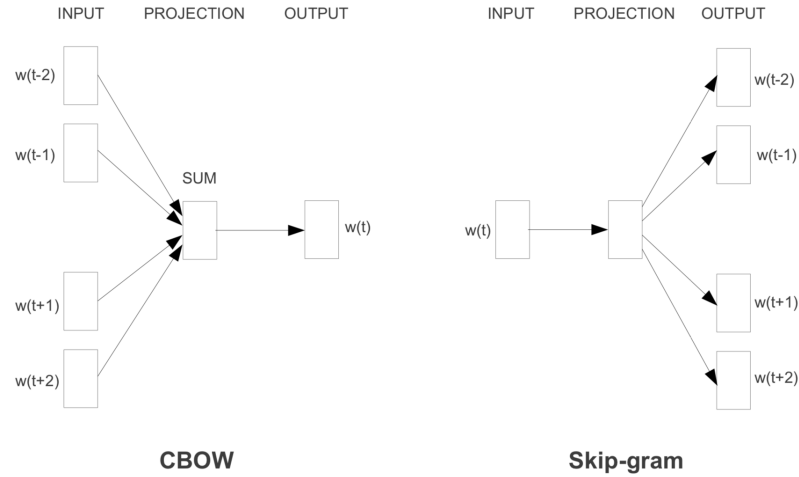


Figure 2.6.1: Illustration of word2vec architectures [177]

2.6.2 GloVe

Global Vectors (or GloVe) for short, is a text embedding algorithm that not only makes use of local statistics, similar to word2vec, but also employs global count statistics. The problem with word2vec that GloVe tries to solve is similar to the problem that tf-idf solves for word count bag of words. Word2vec only takes local contexts into account. For example, the words “the” and “man” might be used together often, but word2vec cannot differentiate whether this is because “the” is a common word or if it is because “the” and “man” have a strong intrinsic relationship. Using global count statistics, i.e. counts across the entire corpora, can help alleviate this issue. GloVe builds a co-occurrence matrix of the words in the corpora. It learns word representations using a neural network with the training objective being to predict the co-occurrence ratios.

2.6.3 Paragraph2Vec

Paragraph2vec is an extension of the word2vec algorithms which can be used to infer meaningful vectors for variable-length pieces of texts - sentences and documents. Here, variable length texts are denoted as *paragraphs*. Every paragraph is mapped to a unique vector, and every word in the paragraph

is mapped to a word vector. Similar to word2vec, paragraph2vec has two forms, namely, the Distributed Memory Model of Paragraph Vectors (PV-DM) and the Distributed Bag of Words version of Paragraph Vector (PV-DBOW).

Distributed Memory Model of Paragraph Vectors (PV-DM)

PV-DM is an extension of the CBOW word2vec algorithm. Each paragraph in the corpus is represented by a paragraph vector d . These paragraph vectors are randomly initialized before training. Each word in a paragraph is represented by a word vector w . For each paragraph, similar to CBOW, we aim to predict a word given its surrounding context. The paragraph vector and word vectors are concatenated to predict the next word in a context. The contexts are fixed-length and sampled from a sliding window over the paragraph. Figure 2.6.2 shows an illustration of the algorithm.

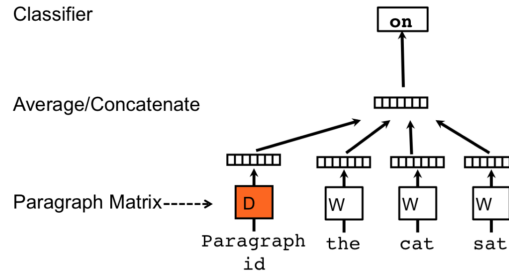


Figure 2.6.2: Illustration of the Distributed Memory Model of Paragraph Vectors (PV-DM) [147]

Distributed Bag of Words version of Paragraph Vector (PV-DBOW)

PV-DBOW is an extension to the skipgram word2vec algorithm. PV-DM considers the concatenation of the paragraph vector with the word vectors to predict the next word in a context window. PV-DBOW, on the other hand, ignores the context words in the input, and instead forces the model to predict words randomly sampled from the paragraph in the output. In more detail, at each iteration of stochastic gradient descent, we first sample a text window. Next, we sample a word at random from the text window. We then train the network on the task of predicting the words in the text window using the paragraph vector and word vector of the one sampled word. Figure 2.6.3 shows an illustration of the algorithm.

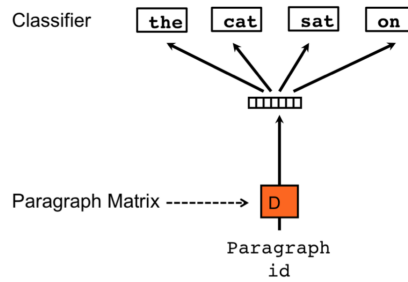


Figure 2.6.3: Illustration of the Distributed Bag of Words version of Paragraph Vector (PV-DBOW) [147]

2.7 Summary

This chapter gave a summary of the theoretical background underpinning our research. As we aim to wade through textual Twitter data in order to evaluate its utility for syndromic surveillance, our work is related to the field of text mining as well as that of natural language processing to aid in the manipulation and understanding of our data. The core scientific field of our work, however, is defined by the branch of artificial intelligence known as machine learning, which aims to automatically discover patterns within data that can be used to inform or make decisions. Particularly, supervised and semi-supervised learning represents the main focus of our work. The problem in such scenarios is formulated as follows: Given a set of input data X and corresponding target responses or outputs, y , we seek to discover a function f which maps from X to y . In our work, we make use of classification algorithms such as bayesian modelling and support vector machines as well as self-training and graph-based learning. As explained in this chapter, such approaches require some degree of manual feature engineering. For this reason, we also investigated deep learning approaches. In addition, we incorporated basic ideas and practices from natural language processing such as stemming, lemmatization and forms of the term-document matrix and word-context matrix. The theoretical ideas and notions in this chapter inform the work in subsequent chapters.

Chapter 3

A Scoping Review of the use of Twitter for Public Health Research

3.1 Introduction

Surveillance, described by the World Health Organisation (WHO) as “the cornerstone of public health security” [281], is aimed at the detection of elevated disease and death rates, implementation of control measures and reporting to the WHO of any event that may constitute a public health emergency or international concern. Syndromic surveillance can be described as the real-time (or near real-time) collection, analysis, interpretation, and dissemination of health-related data, to enable the early identification of the impact (or absence of impact) of potential human or veterinary public health threats that require effective public health action [256]. The task of syndromic surveillance is an undertaking motivated by the notion of public health. Public health has been defined as the science and art of preventing disease, prolonging life and promoting human health through organized efforts and informed choices of society, organizations, public and private, communities and individuals [277]. In this sense, the concept of health encompasses the physical, emotional and social well-being. Historically, public health practitioners have used data from multiple sources for measuring the burden of diseases and other health outcomes, preventing and controlling diseases and guiding healthcare activities. Emergency department attendances or general practitioner (GP, family doctor) consultations are some of the sources traditionally used to track specific syndromes such as influenza-like illnesses (ILI). With the proliferation of the internet and the advent of modern technology, potential new data sources present themselves. In re-

cent years, researchers have recognized that social media platforms, such as Twitter and Facebook, could also provide data about national-level health and behaviour [189].

Among these social media platforms, Twitter offers a unique and potentially powerful data source due to its ease of access, real-time nature and richness in detail. In this paper, we look towards Twitter with the aim of investigating and assessing its utility as a public health tool by performing a scoping review on the subject. While we seek to review the literature of Public health research making use of Twitter, our interest in such literature is limited to research concerning the monitoring, detection and forecasting of public health conditions. We are not interested in social science research investigating the use of Twitter for recruitment or public awareness and dissemination of public health information. We are similarly not interested in research concerned with opinion mining to understand public opinion on public health issues.

A scoping review such as ours is pertinent as there exist no broad and recent evidence-reviews on the use of Twitter data for health research purposes. Wargon et al. [271] performed a systematic review on syndromic surveillance models used in forecasting emergency department visits, however, only 9 studies were found and none of them made use of Twitter or any social media. Subsequently, Charles-Smithe et al. [38] carried out a systematic review of the use of social media (not limited to Twitter) specifically for disease surveillance and outbreak management. Sinnenberg et al. performed another systematic review looking at Twitter as a tool for health research [243]. Their systematic review encompassed research in both the sciences and social sciences. We seek to carry out a scoping review in order to map the broad area of Twitter for public health research as well as to produce an updated review containing more recent studies carried out since the above reviews were published.

3.2 Method

A scoping review methodology was chosen to achieve our goal of mapping the state of Twitter applications in the field of public health research. The scoping review is defined by Arksey and O'Malley [11] as a study that aims "to map rapidly the key concepts underpinning a research area and the main sources and types of evidence available, and can be undertaken as stand-alone projects in their own right, especially where an area is complex". For our scoping review, we made use of the Arksey and O'Malley framework which adopts a rigorous process of transparency, enabling replication of the search strategy and increasing the reliability of the study findings.

3.2.1 Search Strategy

To gain a broad coverage of the available literature, the general terms “*Twitter*” and “*Public Health*” were used as search keywords. We chose these two keywords as “*Twitter*” covers every discussion of the Twitter platform, and used together with “*Public Health*” covers all mention of Twitter in a health context. As our work is multidisciplinary in that it spans multiple fields, we conducted our search in both health and Information Technology (IT) databases. First, we performed a literature search in the health/medical database PubMed. Next, we searched the IT databases IEEE Xplore and the ACM Digital Library. Finally, we searched a general database that indexed both fields, Scopus. Our searches were refined such that we only included research articles which were peer-reviewed and in English. We also limited our search to only return results within the date range of January 2009 and March 2019, which was when the search was carried out. We started our search from 2009 because of the highly influential Google Flu Trends paper published that year which inspired and kickstarted the use of social media as a data source for public health research [85].

3.2.2 Study Selection

754 research articles were returned by our search and 1 paper was added from the bibliographic listings of relevant retrieved papers. Of these 755 articles, we found 550 to be unique. We then drew up a list of criteria for inclusion and exclusion of articles in our review similar to those used by Shatte et al [239]. These criteria are shown in table 3.1. In short, articles were included if all the following criteria were met: (i) the article reported on a method or application of Twitter data to address a public health issue; (ii) the article evaluated the performance of the statistical or machine learning technique used in drawing utility from the Twitter data; (iii) the article was published in a peer-reviewed publication and (iv) the article was available in English. Articles were excluded if any of the following criteria were met: (i) the article did not report an original contribution (e.g. review papers or articles commenting or speculating on the state or future of such research); (ii) the article was focused on the use of Twitter for public health in the context of recruitment and outreach, public awareness and communication, information dissemination or opinion mining; (iii) the article did not make known the statistical or machine learning technique being used; (iv) the full text of the article was not available (e.g. conference abstracts). Guided by our inclusion and exclusion criteria, we identified and selected 92 articles to be included for the review.

Criterion	Inclusion	Exclusion
Time period	2009 - 2019	Studies outside these dates
Language	English	Non-english articles
Article Type	Original peer-reviewed research	Research that was not peer-reviewed
Literature focus	<ul style="list-style-type: none"> Articles reporting on a method or application of Twitter data to address a public health issue. Articles which evaluated the performance of the statistical or machine learning technique used in drawing utility from the Twitter data. 	<ul style="list-style-type: none"> Review articles and other articles not reporting an original contribution. Articles not focused on our above definition of public health but rather concerned with public health in the context of recruitment and outreach, public awareness and communication, information dissemination or opinion mining. Articles which do not make known the statistical or machine learning technique being used. Articles which are works in progress or otherwise do not contain the full-text, such as conference abstracts.

Table 3.1: Inclusion and exclusion criteria.

3.2.3 Information extraction and analysis plan

The focus of our review was to get an exploratory map of the key problems and concepts being tackled in the public health space through the use of Twitter and the techniques being used. To this effect, for each article in our review, data was collected on (i) the aim of the research (ii) the disease or illness of focus (iii) sources of data for the study (iv) statistical or machine learning algorithms and methods used (v) the country for which the study was carried out (vi) the year in which the study was carried out. To analyse the collected information, we used a narrative review synthesis to capture the broad range of research studying Twitter for public health in our scoping review.

flowchart [180]. The mode publication year for articles was 2017 with a range of 2011 - 2019. 19 countries were represented in the studies, with the top 5 countries being the *United States of America (US)*, *United Kingdom (UK)*, *Canada*, *India* and *China*. See fig 3.3.2 for a breakdown of study activity by country. The use of Twitter data was evident for a varied

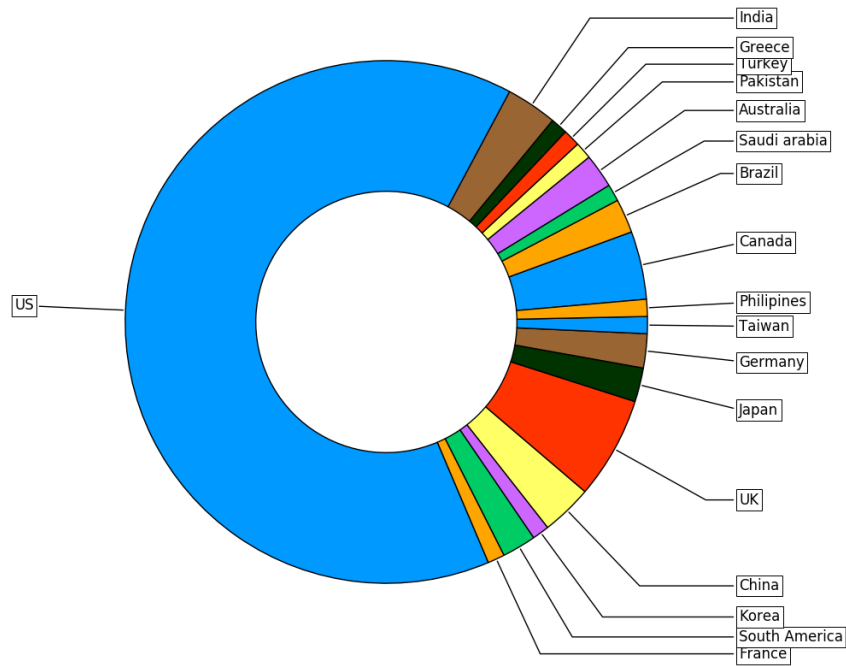
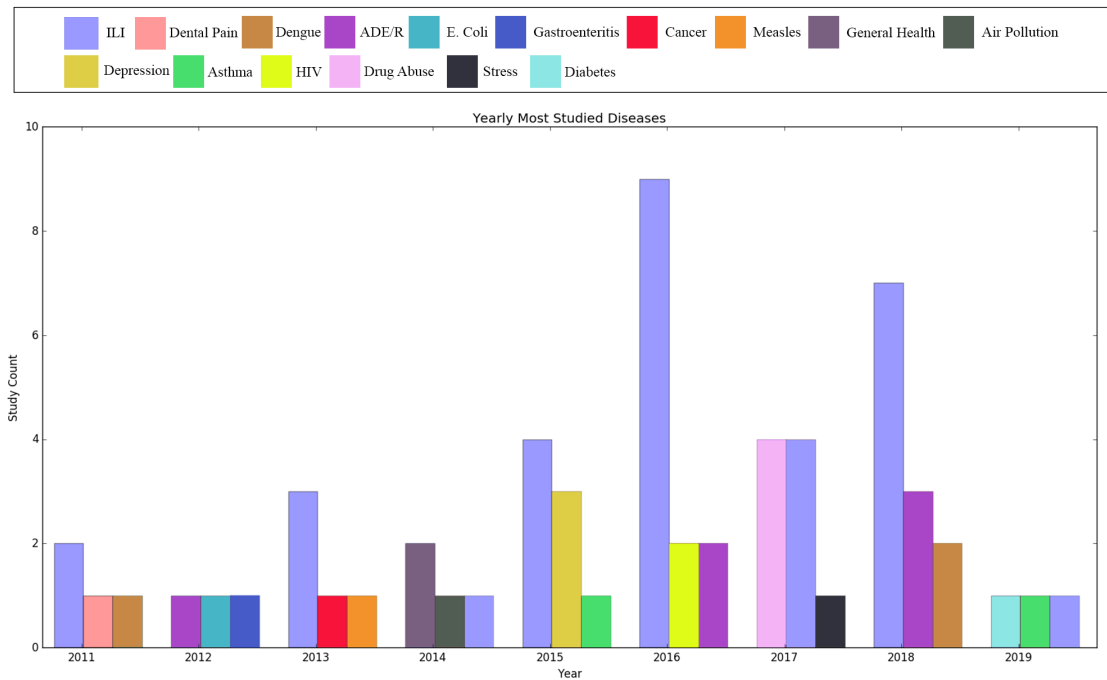


Figure 3.3.2: Breakdown of studies by country

number of different diseases and health conditions. We observed a range of applications dealing with *physical health and illnesses* ($n = 82$) [e.g. influenza-like illnesses (ILIs), adverse drug events and reactions, sexually transmitted diseases, food-borne illnesses], *mental health* ($n = 6$) [e.g. suicide and depression], *natural disasters and environmental issues* ($n = 5$) [e.g. earthquakes, heat waves, air pollution] and *social issues* ($n = 8$) [e.g. drug abuse, smoking, alcoholism]. We examined the subjects of the studies for trends in Twitter applications. We analyzed and plotted the three most studied diseases for each year. Fig 3.3.3 shows the result of this analysis. Taking a closer look at the diseases, conditions and public health phenomena studied using Twitter data, we observed ILIs to be the most common. The next most common subject of public health research using Twitter were drug abuse and adverse drug events and/or reactions (ADE/R). Furthermore, we observed a general rise in the quantity of research into the use of Twitter for public health. Research activity appears to have peaked in 2016 but seems to be on the rise from 2018. As this scoping review looks at studies up until March 2019, the data for 2019 is incomplete. This limitation is due to the fact that this review can only investigate studies until the time of its writing, which happened to be early in the year. A myriad of statistical and machine learning techniques were used in the analysis of Twitter data for public health. Most studies implemented

Figure 3.3.3: Most studied diseases each year.¹

just one technique ($n = 54$) but some others made use of a mix of methods and techniques ($n = 38$). The articles made use of a range of statistical and machine learning techniques including *supervised learning* ($n = 70$) [e.g. Support Vector Machine (SVM), naive bayes, decision trees, logistic regression], *unsupervised learning* ($n = 18$) [e.g. clustering, association rule mining], *semi-supervised learning* ($n = 4$) [e.g. graph learning, transductive support vector machine (t-SVM)], *text analysis and natural language processing* ($n = 23$) [e.g. latent Dirichlet allocation (LDA), biterm topic modelling, lexicon analysis], *deep learning* ($n = 16$) [e.g. Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), word and document embeddings], *statistical modelling and analysis* ($n = 12$) [e.g. correlation analysis, partial differential equation (PDE), TRAP] and *time series analysis* ($n = 7$) [e.g. Autoregressive Integrated Moving Average (ARIMA), time-series Susceptible-Infected-Recovered (TSIR) model]. The average number of Tweets used in the reviewed studies was roughly twenty thousand. A closer look at the research towards Twitter use for public health revealed that the SVM was a popular tool in this research field. We hypothesize that this is due to the SVM's popularity and strength in text classification problems [120]. We also analyzed the surveyed studies to find out which statistical or machine learning algorithms were popular, as well as if and how this might have shifted over time. Fig 3.3.4 shows a plot of the most used algorithms for each year covered in this review. Lexicon-based analysis proved popular between 2012 until 2014. After this, Bayesian learning seemed to be the method of choice, followed by the SVM. From 2018, the widespread popularity of deep learning appears to have made its

way into public health research with Twitter data, as it is becoming the dominant method used since then.

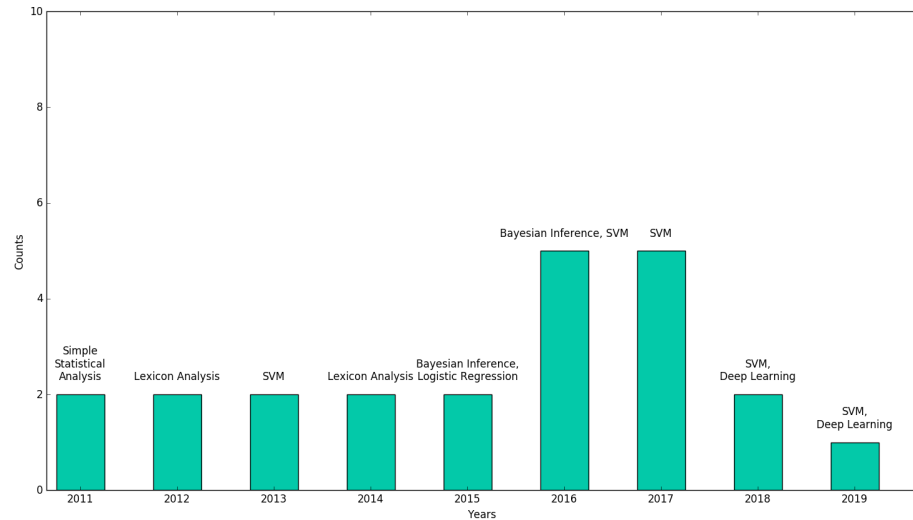


Figure 3.3.4: Most applied algorithms each year

3.3.2 Application Domains of Twitter in Public Health

Through the synthesis of the data obtained from the reviewed articles, we broadly identified 6 different ways in which Twitter data is used for public health research. The identified domains were: (i) *surveillance* ($n = 41$); (ii) *event detection* ($n = 38$); (iii) *pharmacovigilance* ($n = 19$); (iv) *forecasting* ($n = 15$); (v) *disease tracking* ($n = 12$) and (vi) *geographic identification* ($n = 7$). *Surveillance* includes articles aiming to monitor some status over a period of time. *Event detection* includes articles that aim to discover and/or identify a health-related event from Twitter data. *Pharmacovigilance* includes articles which were concerned with public drug consumption and reactions to said drugs. *Forecasting* includes articles which aim to predict the trends for health-related events. *Disease tracking* includes articles attempting to observe or predict the spread of diseases in the public through Twitter. *Geographic identification* includes articles whose aim is to geolocate Twitter users, usually in order to facilitate or improve the application of one of the other domains.

We were interested in examining the trends, if any, in the public health application domains studied over the years. We constructed a bubble trend chart from the reviewed papers. This chart, included in fig 3.3.5, illustrates the research activity in each domain for each year. It shows that there appears to indeed be a trend in activity for different public health domains.

In 2011, there is little to moderate activity across the board. In the years following that, we see research in some domains drop off and on the map, and some growing steadily in size. Event detection, surveillance and pharmacovigilance appear to have seen steady increases in activity, leading the other domains. However, since 2016, research in those three domains has reduced slightly, with some focus switching to the other domains. The data for the year 2019 is not particularly informative, as the scoping review was only carried out in the first quarter of 2019.

We were also interested in the different techniques applied across different

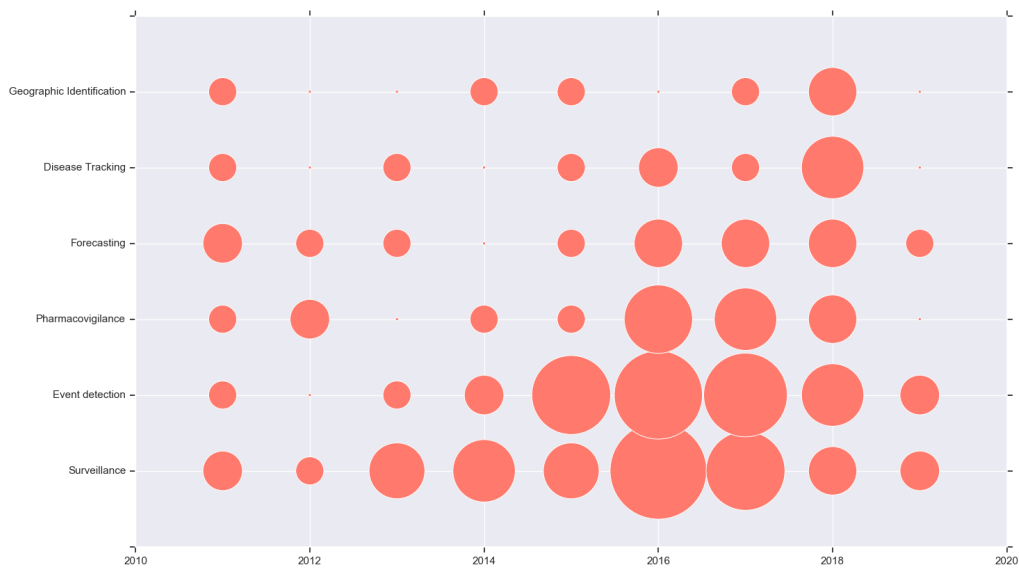


Figure 3.3.5: Bubble chart showing the trends of research activity in public health application domains with time.

public health research domains. We computed a matrix of the application domains against the techniques applied and visualised it as a heatmap. This heatmap is shown in fig 3.3.6. Darker colours in the heatmap indicate higher activity for that cell. Supervised learning appears to see a lot of utility across the board. Deep learning and natural language processing also see a fair amount of utility, particularly in event detection, pharmacovigilance and surveillance. Unsupervised learning seems to see some utility use in surveillance and event detection. On the other hand, semi-supervised learning appears to see the least use across the board.

The reviewed articles were found to exist within one or more of these domains. These domains are discussed in more detail below.

¹Note that the information shown for 2019 is not comparable to that for other years due to the fact that, at the time of plotting the graph, 2019 had not elapsed.

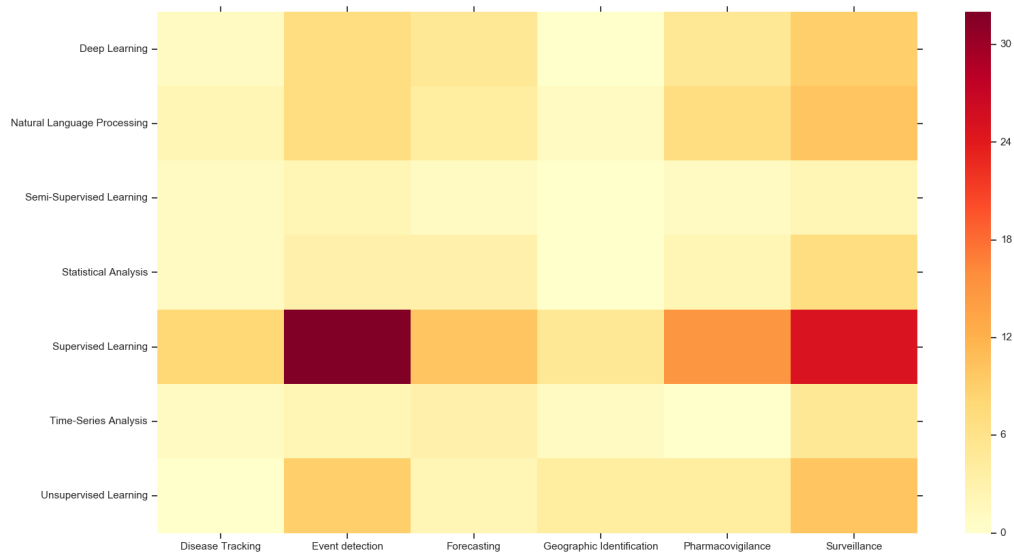


Figure 3.3.6: Most applied algorithms each year

Surveillance

Surveillance was the most popular research domain with around 43% of the reviewed articles represented. Research on surveillance focused on employing machine learning in order to utilize Twitter as an alternative or augmentative resource to traditional health surveillance systems. Naturally, the surveillance domain encompasses the field of syndromic surveillance [93, 40, 115]. However, it is broad and also includes additional applications such as the tracking of vaccination efforts [247] and monitoring of environmental conditions [107, 123], as well as for natural disaster reporting and alarming [13]. That being said, the most common application was the syndromic surveillance of influenza-like illnesses (ILIs). Besides ILIs, other diseases and conditions that were studied include dengue, HIV, gastroenteritis, ebola, diarrhoea and allergies. Due to the extensive research carried out in this area, a wide range of techniques were used. For example, supervised learning applied in the form of k-Nearest Neighbours (kNN) was used to monitor allergy trends and occurrences [187]. Unsupervised learning was used in the form of Density-based Spatial Clustering of Applications with Noise (DBSCAN) clustering in order to exploit the spatial and temporal properties of the Twitter stream for dengue surveillance [86]. Semi-supervised learning was used in the form of transductive SVMs for the surveillance of ILIs, gastroenteritis, diarrhoea and vomiting [254].

Table 3.2: Summary of statistical and machine learning methods and data sources for surveillance using Twitter data

Public Health Issue	Method	Comparative Data Source
Cancer	Simple Statistical Analysis [150]	CDC
Hepatitis A	Support Vector Machine [140]	
Gastrointestinal Illnesses	Correlation Analysis [132]	Government of ontario, Kingston, Frontenac and Lennox & Addington Public Health
Suicide	ARIMA (Autoregressive Integrated Moving Average [173]	
HIV	Graph Modelling [253], Word2Vec [28], Doc2Vec [28], Dynamic Topic Modeling [28]	
Allergies	K-Nearest Neighbour [187], Bayesian Inference [187], Support Vector Machine [187]	
Heat Wave	Near Regression [123], ARIMA (Autoregressive Integrated Moving Average) [123]	The US National Oceanic and Atmospheric Administration (NOAA) National Centers for Environmental Information (NCEI)
Heat Related Illnesses	Correlation Analysis [132]	Government of ontario, Kingston, Frontenac and Lennox & Addington Public Health
Depression	ARIMA (Autoregressive Integrated Moving Average [173]	
Syphilis	Binomial Regressions [289]	CDC
Ebola	Bayesian Inference [195], Lexicon Analysis [195]	

Respiratory Illnesses	Correlation Analysis [132]	Government of Ontario, Kingston, Frontenac and Lennox & Addington Public Health
E Coli	Latent Dirichlet Allocation [73], Lexicon Analysis [73]	Robert Koch Institute
Measles	Support Vector Machine [140]	
Influenza-like Illnesses (Hemophilus)	Bayesian Inference [115]	Genbank
Vomiting	TSVM [254], ARIMA (Autoregressive Integrated Moving Average) [254]	Public Health England
Gastroenteritis	TSVM [254], Latent Dirichlet Allocation [73], Lexicon Analysis [73], ARIMA (Autoregressive Integrated Moving Average) [254]	Public Health England, Robert Koch Institute
Salmonella	Support Vector Machine [140]	
Food Borne Illness	Support Vector Machine [225]	Southern Nevada Health District (SNHD)
Earthquake	Clustering [13], Bayesian Inference [13]	
Stress	Ordinal Regression [159]	
Air Pollution	Self-Organizing Map (Clustering) [220], Cross-Correlation [107]	The European Centre for Medium-Range Weather Forecasts (ECMWF), London Air Quality Network

Influenze-like Illnesses (ILI)	Lexicon Analysis [156], Deep Learning (CNN) [296], Fp-Growth [201], Bayesian Inference [238],[109], Correlation Analysis [132], Deep Learning (RNN) [296], Deep Learning (MLP) [152], Fasttext [296], Bayesian Inference [34],[156], ARIMA (Autoregressive Integrated Moving Average) [254],[31], Simple Statistical Analysis [150], Support Vector Machine [6],[201], Glove [296], Maximum Entropy [34], TSVM [254], Partial Differential Equation [266], Autoregressive Moving Average (Arma) [2], Outlier Detection [60], Topic Model [41], Temporal Topic Model [40], Logistic Regression [31], Count Correlation [247]	Public Health England, Frontenac and Lennox & Addington Public Health, Chinese CDC, Pan American Health Organization (PAHO), CDC, HHS data, Kingston, FluWatch, Government of ontario, The Pan American Health Organization (PAHO)
General Health ¹	Topic Model (Ailment Topic Aspect Model (Atam)) [200], Lexicon Analysis [56], Regression [56], Simple Statistical Analysis [288], Temporal Ailment Topic Aspect Model (TM-ATAM) [241]	CDC, U.S. Census' State-Based Counties Gazetteer
Dengue	Dbscan (Clustering) [86], Deep Learning (RNN) [160], Word Embeddings (Glove) [160], Simple Statistical Analysis [66]	Brazilian Health Ministry, Philippine's Department of Health, Brazilian Official Dengue case data

¹Generic feelings of unwellness and non-specific illness

Diarrhoea	TSVM [254], ARIMA (Autoregressive Integrated Moving Average) [254]	Public Health England
Obesity	Dbscan (Clustering) [133]	

Event Detection

Detection was another popular domain which saw around 40% of the reviewed articles represented. Research in this domain seeks to automatically discover events and describe the magnitude and trend of disease, as well as the impact of control measures. Event detection differs from surveillance in that surveillance is an activity of vigilance, with the aim of finding things before they become full-blown events. Event detection aims to find existing full-blown occurrences. Research in this domain seeks to automatically detect events and describe the magnitude and trend of disease, as well as the impact of control measures. Examples of applications in this domain are automatically detecting drug abuse within the population [166], depression and suicide [173], ebola [224] and most common of all, ILI [59]. Such research tends to be fairly recent with the mode publication year being 2016. The statistical and machine learning techniques used were typically supervised, with most studies employing either classification or regression to make the predictions necessary for detection. For example, SVMs were used to detect mention of “dabbing”, a method of marijuana consumption that involves inhaling vapors from heating marijuana concentrates [83]. CNNs were used to detect harmful algal blooms from pictures posted on Twitter [143]. Additionally, stepwise regression was used to detect depression from Tweets in order to explore the effect of climate and seasonality on mood [286].

Table 3.3: Summary of statistical and machine learning methods and data sources for event detection using Twitter data

Public Health Issue	Method	Complementary Data
Cancer	Support Vector Machine [78]	CDC
Smoking	Bayesian Logistic Regression [9]	
Suicide	ARIMA (Autoregressive Integrated Moving Average) [173]	

Harmful Algal Blooms (HABS)	Deep Learning (CNN) [143]	
HIV	Decision Tree [3], Support Vector Machine [3], Graph Modelling [253], Multilayer Perceptron [3]	
Allergies	[151], Bayesian Inference [151]	pollen.com, National Climatic Data Center Climate Data Online (CDO)
Drug Abuse	Biterm Topic Model [166], Decision Tree [210], Support Vector Machine [83], Topic Model [167]	
HPV	Decision Tree [172], Linear Classifier [172]	
Infectious Intestinal Diseases (IID)	Word2Vec [295], Gaussian Process [295]	Public Health England
Adverse Drug Events (ADE)	Multi-Instance Logistic Regression [269]	
Depression	Non-Negative Matrix Factorization [285], ARIMA (Autoregressive Integrated Moving Average) [173], Simple Statistical Analysis [186], Stepwise Regression [286]	National Climatic Data Center, National Oceanic and Atmospheric Administration (NOAA)
Ebola	Lexicon Analysis [224], Support Vector Machine [224]	
Back Pain	Logistic Regression [149]	
Vomiting	TSVM [254], ARIMA (Autoregressive Integrated Moving Average) [254]	Public Health England
Gastroenteritis	TSVM [254], ARIMA (Autoregressive Integrated Moving Average) [254]	Public Health England
Asthma	Support Vector Machine [78]	CDC

Food Borne Illness	K-Nearest Neighbour [98], Support Vector Machine [225]	Southern Nevada Health District (SNHD), CDC
Earthquake	Clustering [13], Bayesian Inference [13]	
Diabetes	Support Vector Machine [78]	CDC
Dental Pain	Simple Statistical Analysis [100]	
Influenza-like Illnesses (ILIs)	Clustering [61], Lexicon Analysis [59],[154],[156], Deep Learning (RNN) [296], Logistic Regression [30], Gaussian Process [264], Deep Learning (CNN) [296], Outlier Detection [60], Bayesian Inference [59],[156], Fast-text [296], ARIMA (Autoregressive Integrated Moving Average) [254], GloVe [296], FP-Growth [201], Trap Model [265], Support Vector Machine [201],[279],[30], Shallow MLP [108], TSVM [254], Word2Vec [61], Regression [279]	Penn State's Health Services, Infectious Disease Surveillance Center, Royal College of General Practitioners (RCGP), Public Health England, CDC
General Health ¹	Support Vector Machine [144], Lexicon Analysis [144]	
Diarrhoea	TSVM [254], ARIMA (Autoregressive Integrated Moving Average) [254]	Public Health England
Obesity	Dbscan (Clustering) [133]	
Middle East Respiratory Syndrome (Mers)	Lexicon Analysis [224], Support Vector Machine [224]	

¹Generic feelings of unwellness and non-specific illness

Pharmacovigilance

Research in pharmacovigilance focused mainly on adverse drug reactions and events, but also investigated with recreational drug use and abuse. Pharmacovigilance involves monitoring Twitter for signs of ill or unintended effects and side-effects of pharmaceutical products. Usually, when studying the use of Twitter to detect adverse drug reactions and events, articles searched for a range of names obtained from a thesaurus of drugs and events, such as the Medline Plus Drug Information [204]. However, other such studies focused on a drug for a particular disease such as HIV [3]. In addition, studies also investigated drug habits and their effects on the population. For example, one article studied the use of e-cigarettes and their utility for smoking cessation [9]. Another article studied the variability of alcoholism with time [273]. A number of the pharmacovigilance studies utilized sentiment analysis, usually a form of supervised text classification, to aid in their efforts [3, 204, 28]. In fact, most of the studies make use of supervised learning in the form of text classification using mostly SVMs and decision trees. Of the 19 articles in this domain, three made use of deep learning [28, 157, 91], one employed a semi-supervised multi-instance learning approach [91] and three used unsupervised natural language processing [28, 127, 167].

Table 3.4: Summary of statistical and machine learning methods and data sources for pharmacovigilance using Twitter data

Public Health Issue	Method	Complementary Data
Smoking	Bayesian Logistic Regression [9]	
HIV	Support Vector Machine [3], Word2Vec [28], Doc2Vec [28], Multilayer Perceptron [3], Decision Tree [3], Dynamic Topic Modeling [28]	
Vaccination	Semantic Network Analysis [127]	
Drug Abuse	Decision Tree [210], Support Vector Machine [83], Topic Model [167], Simple Statistical Analysis [39]	National Surveys on Drug Usage and Health (NSDUH)

Adverse Drug Reactions (ADRs)	Conditional Field [137],[157], Lexicon Analysis [137],[196], Deep Learning (RNN) [91], Word Embeddings (Glove) [91], Word2Vec [157]	ADRMine
Adverse Drug Events (ADEs)	Multi-Instance Logistic Regression (Milr) [269], Semi-Supervised Multi-Instance (Nssm) [268], Bayesian Inference [204], Support Vector Machine [204],[19], Lexicon Analysis [19]	
Alcoholism	Simple Statistical Analysis [273]	
Miscellaneous	Decision Tree [92], Support Vector Machine [126], Latent Dirichlet Allocation [126]	

Forecasting

Forecasting research studies the prediction of public health trends, as well as means of *nowcasting* which is the prediction of the present state of public health. It can be seen as a part of the syndromic surveillance effort, aimed at predicting epidemics in order to improve crisis response. Research in this domain is focused predominantly on ILIs. Around 67% of the reviewed literature studied ILI. However, other diseases such as dengue, gastroenteritis, cancer and asthma were also studied [66, 254, 150, 216]. While a mix of statistics and machine learning is used in this domain, there is a heavier focus on statistics. In fact most studies made use of statistical techniques like regression and time series analysis. For example, dynamic regression was used to predict influenza trends in Boston, USA [164]. AutoRegressive Integrated Moving Average (ARIMA) was used to forecast influenza cases on a city level in Chongqing, China, as well as for predicting gastroenteritis in the UK [251, 254]. Partial differential equations were used to forecast influenza cases on a regional level across the USA [266]. Deep learning was also used to aid in the forecasting problem of predicting influenza cases [152] and in the creation of SENTINEL, a software system capable of nowcasting diseases being monitored by the US Centre for Disease Control (CDC) [236]. Unsupervised learning was used in the form of topic

modelling in a study aiming to predict health transition trends without any *a priori* diseases [241].

Table 3.5: Summary of statistical and machine learning methods and data sources for forecasting using Twitter data

Public Health Issue	Method	Complementary Data
Cancer	Simple Statistical Analysis [150], Linear Regression [262]	CDC
E Coli	Latent Dirichlet Allocation [73], Lexicon Analysis [73]	Robert Koch Institute
Vomiting	TSVM [254], ARIMA (Autoregressive Integrated Moving Average) [254]	Public Health England
Gastroenteritis	TSVM [254], Latent Dirichlet Allocation [73], Lexicon Analysis [73], ARIMA (Autoregressive Integrated Moving Average) [254]	Public Health England, Robert Koch Institute
Asthma	Decision Tree [216], Shallow MLP [216]	Children's Medical Center (CMC)
Influenze-like Illnesses (H1N1)	Support Vector Regression [242]	CDC
Influenze-like Illnesses	Deep Learning (RNN) [296], Deep Learning (MLP) [152], Fasttext [296], Deep Learning (CNN) [296], ARIMA (Autoregressive Integrated Moving Average) [254],[251], GloVe [296], Temporal Topic Model [40], Dynamic Regression [164], TSVM [254], Partial Differential Equation [266], Simple Statistical Analysis [150], Autoregressive Moving Average (ARMA) [2]	Boston Public Health Commission, Public Health England, Pan American Health Organization (PAHO), Chinese CDC, CDC

General Health ¹	Temporal Ailment Topic Aspect Model (TM-ATAM) [241]	CDC
Dengue	Simple Statistical Analysis [66]	Brazilian Official Dengue case data
Diarrhoea	TSVM [254], ARIMA (Autoregressive Integrated Moving Average) [254]	Public Health Eng- land

Disease tracking

Disease tracking is a domain that seeks to support epidemiology by offering insight into the spread of infectious diseases. Research in this domain is primarily interested in understanding the way in which diseases spread through a population. It looks toward not only gaining a better understanding of the spread of diseases, but also to keep track of the public health state during recognized outbreaks and mass gatherings which could be a breeding ground for disease. For example, one study investigated and proposed a means of tracking flu transmission in China using Twitter [109]. Another study retrospectively tracked the spread of measles during the 2015 outbreak [252]. Additionally, there was a study to detect the occurrence and spread of disease symptoms which could signify a potential outbreak at a number of British music festivals and a religious event in Mecca, Saudi Arabia [288]. Most studies in this domain made use of machine learning methods, leaning towards supervised learning. In particular, regression learning proved popular, as two studies utilized dynamic regression and support vector regression to track the spread of influenza [164, 242]. Another study proposed a gaussian mixture regression approach to estimating the geographic origin of a tweet for use during an outbreak [111]. There were also some studies which used statistical analysis to obtain impressive results. One of such studies made use of the TSIR (time-series Susceptible-Infected-Recovered) model to understand human mobility and the spread of the dengue virus in Lahore, Pakistan [138]. While it was rare, one study made use of semi-supervised learning and deep learning to simulate influenza epidemics.

¹Generic feelings of unwellness and non-specific illness

Table 3.6: Summary of statistical and machine learning methods and data sources for disease tracking using Twitter data

Public Health Issue	Method	Complementary Data
Measles	Semantic Network Analysis [252]	CDC
Influenze-like Illnesses (Hemophilus)	Bayesian Inference [115]	Genbank
Influenze-like Illnesses (H1N1)	Semi-Superviseddeep Learning (MLP) [294], Support Vector Regression [242]	CDC
Influenze-like Illnesses	Bayesian Inference [109], Bayesian Inference [34], Dynamic Regression [164], Maximum Entropy [34]	FluWatch, Boston Public Health Commission, Chinese CDC
General Health ¹	Temporal Ailment Topic Aspect Model (TM-ATAM) [241]	CDC
Dengue	Time-Series Susceptible-Infected-Recovered Model [138], Simple Statistical Analysis [66]	Brazilian Official Dengue case data
Miscellaneous	Gaussian Mixture Regression (Gmr) [111]	Map data

Geographic identification

Geographic identification is a small domain which involves the extraction of geographical information from Twitter data and typically sees little use alone. Rather, it is used in conjunction with other domains to improve the efficacy of solutions or provide added benefit. It is most often used with *surveillance* and *disease tracking*. Methods used in geographic identification are typically based on unsupervised learning. For example, DBSCAN clustering was used to monitor and track obesity levels within the population [133], as well as track the spread of the dengue virus [86]. Another study utilized hot spot analysis to examine spatial patterns of depression on Twitter. Some supervised learning, typically in the form of classification is also used in geographic identification. Here, a classifier is used to predict the location of a tweet based on some features of the tweet, usually its word

¹Generic feelings of unwellness and non-specific illness

collocations. As an example, one study in the review made use of a random forest classifier to predict which city and province a tweet determined to be from Canada (according to the Twitter API), was from [230]. While geographic identification in itself is not of major use to the field of public health, when combined with other identified public health research domains, it offers improvements on the specificity and granularity of their results.

Table 3.7: Summary of statistical and machine learning methods and data sources for geographic identification using Twitter data

Public Health Issue	Method	Complementary Data
Depression	Non-Negative Matrix Factorization (Nmf) [285]	
Dengue	Time-Series Susceptible-Infected-Recovered Model [138], Dbscan (Clustering) [86]	Brazilian Health Ministry
Obesity	Dbscan (Clustering) [133]	
Miscellaneous	Latent Dirichlet Allocation [230], Support Vector Machine [230],[116], Bayesian Inference [230], Random Forest [230], Multilayer Perceptron [230], Gaussian Mixture Regression (GMR) [111], HDBSCAN (Clustering) [116]	Map data

Our work looks toward syndromic surveillance. Syndromic surveillance encompasses a number of these applications. Recall from chapter ?? the definition of syndromic surveillance as the real-time (or near real-time) collection, analysis, interpretation, and dissemination of health-related data, to enable the early identification of the impact (or absence of impact) of potential human public health threats. As such syndromic surveillance is not only a surveillance application, but also one of forecasting and event detection. Due to the political nature of syndromic surveillance, as a result of it being carried out for a country or state's population, it can also involve geographic identification in order to delineate boundaries and observation points of interests as needed.

3.4 Discussion

This review aims to canvass the published literature on the use of Twitter data for public health, highlighting popular and current research and applications. Three findings were produced from the review. First, we identified the key application domains being studied: (i) *surveillance*; (ii) *event detection*; (iii) *pharmacovigilance*; (iv) *forecasting*; (v) *disease tracking* and (vi) *geographic identification*. Studies were found to predominantly be concerned with surveillance, event detection and pharmacovigilance. Next, the conditions and diseases being tackled using Twitter data were identified. We discovered a wide range of illnesses to which Twitter data is being applied to including infectious diseases, mental health problems, environmental issues and social issues. Finally, we mapped out the statistical and machine learning algorithms and approaches being used to process and analyse Twitter data for public health purposes. In doing so, we observed trends in these approaches. Bayesian learning and SVMs appear to be popular algorithms of choice, however, in the past two years the focus seems to have shifted towards deep learning.

While research toward using Twitter for public health has been extensive, there exist some gaps for future research to fill. Understandably, studies are focused on infectious diseases. In particular, the reviewed research focused on the surveillance and detection of influenza. There is significant scope to explore whether Twitter data is adequate to study other infectious diseases. We do not expect Twitter data to be of use to the study of sexually transmitted diseases (STDs) as such a study would rely on Twitter user-reporting. Individuals infected with an STD may not be likely to report this on a public platform. That being said, other infectious diseases such as cholera could be studied. Furthermore, it would also be interesting to see if the utility of Twitter and social media for public health extends to non-infectious diseases, such as asthma or celiac disease. Our review did not identify any articles that used Twitter to examine the occurrence of positive health states/outcomes, although this might be a result of the limitations of our scoping methodology.

It was also apparent that the reviewed studies employed a lot of supervised learning techniques. This is somewhat understandable as the most popular application domains were surveillance and detection, and supervised learning deals heavily with classification and prediction. The average number of Tweets used in the reviewed studies was roughly twenty thousand. This suggests that most of the reviewed articles had large amounts of labelled Twitter data available to them. Such large labeled datasets lend themselves well to supervised learning tasks. Unfortunately, such labeling could constitute a sizeable effort. There could be some merit in using Twitter data in an unsupervised manner, for positive outcomes. In fact, both approaches may be combined, making use of Twitter data in both a structured and unstructured way. This is what semi-supervised learning is for. Such approaches would reduce the amount of labeled Twitter data required by also

taking advantage of the unlabeled data. Some articles were already starting to look toward such approaches [268, 294]. However, there are very few of these articles and they are all focused on ILI.

Furthermore, despite the rich potential for success from using Twitter data for public health which was identified in the literature, there were few articles describing active Twitter-based systems and/or their evaluation in an operational context for routine public health practice. This may suggest that it is somewhat difficult to translate research using Twitter for public health into practice. We believe the bulk of this challenge might come from the ethical issues involved and the lack of an ethical framework for the integration of social media into surveillance systems. That being said, public health institutions around the world may already be using Twitter as such a tool, and just not reporting their efforts.

It is also important to note that this review had some limitations. Constraints in the search methodology such as the use of broad search terms and the exclusion of works-in-progress may have resulted in some relevant studies being missed. However, this is a common limitation of scoping reviews as they are intended to broadly map topics, achieving a good balance of breadth and depth in a relatively quick time-frame [209]. As such, this review successfully gives an overview of the state of the field and provides insightful analysis of the existing literature in the field. Such information could be useful in aiding researchers, clinicians and policy makers in understanding the modern landscape of public health applications for social media. To conclude, research into the application of Twitter data for public health has uncovered interesting and inspiring advances, especially in recent years, and identified gaps in the knowledge thus allowing targeted research in the future. Overall, we see that Twitter data can be used to aid in public health efforts concerned with surveillance, event detection, pharmacovigilance, forecasting, disease tracking and geographic identification, demonstrating positive results. With the richness of Twitter as a dataset, together with the development of machine learning tools and their increasing accessibility, we expect to see more interesting ideas and applications of Twitter to public health.

Chapter 4

Working with Twitter Data: Extraction, Preparation and Processing

4.1 Introduction

The focus of this thesis is on the use of Twitter for syndromic surveillance. As such, a large part of our data comes from Twitter. Twitter is a popular free micro-blogging platform. The service allows registered members to publish short posts called “Tweets”. It has around 275 million users [49], with 500 million Tweets posted per day [53]. From this, we can see that large amounts of data are produced by Twitter. This is one of the qualities of the platform that we want to capitalise on. In order to effectively do this, some consideration must be put into the means and manner by which we collect, organise and utilize this data. In this chapter, we discuss the Twitter Application Programmer’s Interface (API) and describe the ways through which we interact with it, as well as the data we obtain. As a starting point, we make use of the free version of the API. We go on to highlight the preprocessing procedures used to prepare the collected raw data for our purposes. We then discuss the transformations and processes used to extract meaningful feature representations from the Twitter data. We present novel means of Tweet feature extraction and representation using emojis

4.2 The Twitter Application Programmer's Interface (API)

Twitter offers 6 main classes of its API.

- (i) Search API
- (ii) Stream API
- (iii) Account Activity API
- (iv) Direct Message API
- (v) Website API
- (vi) Ads API

The search API is used to sample historical Tweets looking back within the past seven days. The stream API is used to collect Tweets in real-time as they are published. The account activity API is used to subscribe to a number of user actions, such as posting a Tweet, or following a user. It makes it possible to keep track of the actions of up to 15 users. The direct message API enables the automation of sending and receiving private messages and can be used to make chatbots. The website API is used for embedding Twitter content within a third-party website. Finally, the ads API is used to create and target advertising campaigns towards Twitter users. Our interest is in the thoughts and feelings of Twitter users. Subsequently, we look to the Tweets posted by users. While the search and streaming APIs make Tweets available to us, the streaming API is more befitting due to the fact that its real-time nature is more appropriate for the task of syndromic surveillance.

We made use the official Twitter streaming API for collecting Tweets. The streaming API provides a subset of the Twitter stream free of charge. The whole stream can be accessed on a commercial basis. Studies have estimated that using the Twitter streaming API, users can expect to receive anywhere from 1% of the tweets to 40% of tweets in near real-time [182]. The streaming API offers some filtering capabilities. It allows the use of up to 400 keywords, used to filter and control the Tweets collected. It also allows filtering by up to 50,000 user IDs and up to 25 geolocation bounding boxes. However, with the free API, it is only possible to choose one route for filtering (i.e. keywords, user IDs or location bounding boxes) at a time. A breakdown of the data contained within a Tweet is shown in figure 4.2.1.

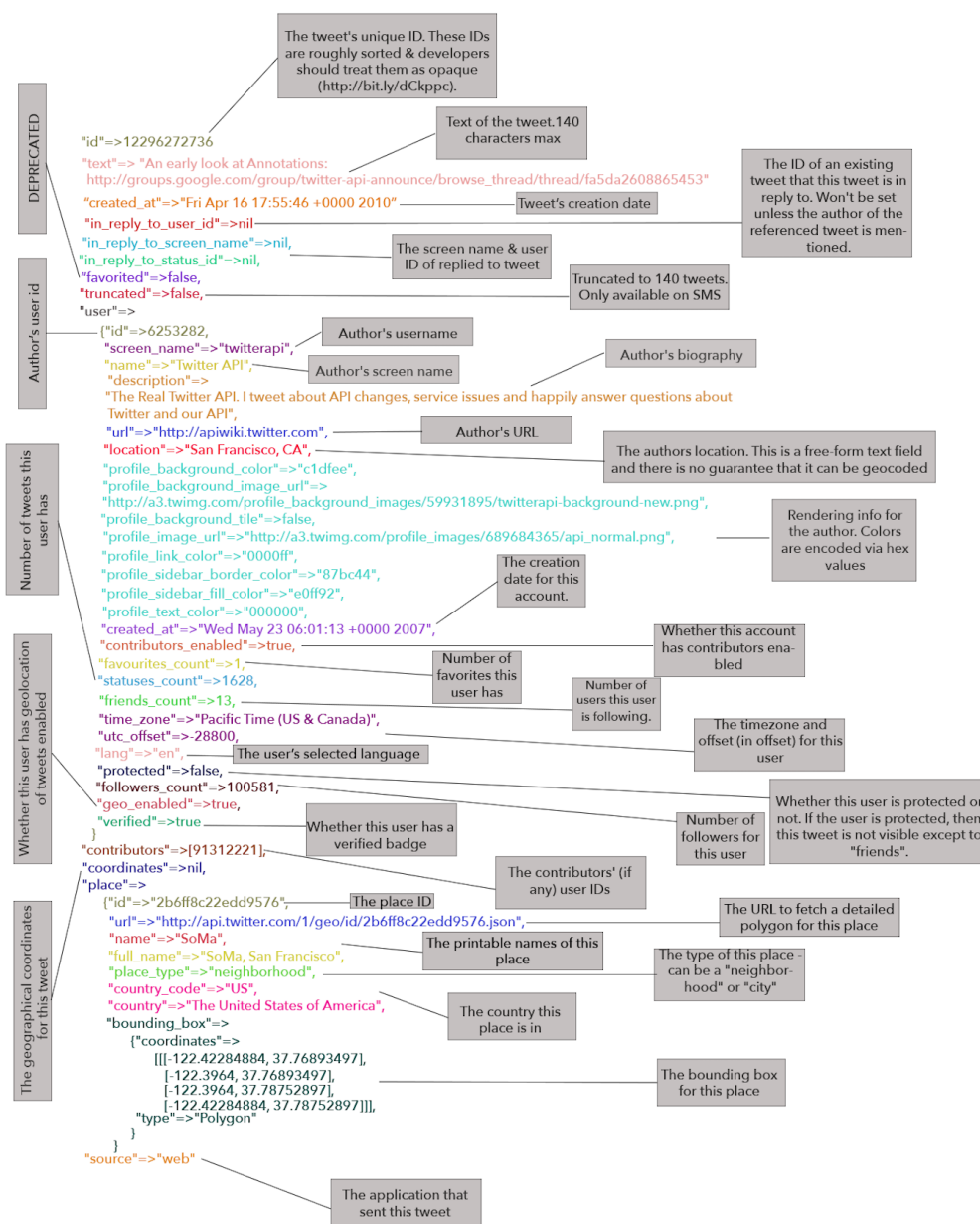


Figure 4.2.1: Map of a Tweet as obtained from the Twitter API

4.3 Data Collection and Preprocessing

For our syndromic surveillance experiments, Tweets were collected in different periods to account for the seasonality of the syndromes under study (i.e. *asthma/difficulty breathing*), and to have a better chance of observing a significant episode, which is largely unpredictable. Observing different periods also enable us to monitor changes in the use of Twitter, as well as changes in the language used on Twitter over time. We started with an Autumn period (September 2015 to November, 2015), followed by a summer period

(June 2016 to August, 2016) and a winter through to mid-summer period (January 2017 to July, 2017). We extracted Tweets in the English language with specific keywords that may be relevant to a particular syndrome. To generate this set of keywords we worked in conjunction with experts from Public Health England (PHE), to create a set of terms that may be connected to the specific syndrome under scrutiny, in this case *asthma/difficulty breathing*. We then expanded on this set using synonyms from regular thesauri, as well as from the urban dictionary¹ as those may capture some of the more colloquial language used by the youth on Twitter. Examples of our keywords are “asthma”, “wheezing”, “couldn’t breathe” etc. The full list of keywords is included in the appendices.

Filtering by keyword can be implemented through the Twitter API by using the provided “track” parameter, followed by a comma-separated list of phrases which will be used to determine which Tweets will be delivered from the real-time stream. A phrase may be one or more terms separated by spaces, and a phrase will match if **all** of the terms in the phrase are present in the tweet, regardless of order and case. Hence, in the API, commas act as logical ORs, and spaces are equivalent to logical ANDs. The tracked terms are matched against a number of attributes of the tweet including the *text* of the Tweet, *expanded_url* and *display_url* of links and media in the Tweet, and *screen_name* of the user.

10 million Tweets were retrieved over the three collection periods. The general characteristics of the collected tweets are reported in Table 4.1. There are several attributes associated with a Tweet which are available to our analysis and they can be seen in figure 4.2.1. Not all of the available attributes are useful for our experiments. As such, we collected only those that could help us in our task. We collected “Tweet.Id”, “text”, “created_at”, “user_id”, “source” as well as information that may help us establish location such as “coordinates”, “time_zone” and “place.country”. We stored the collected tweets using MongoDB², which is an open source no-SQL database whose associative document-store architecture is well suited to the easy storage of the JSON Twitter responses.

4.3.1 Location Filtering

Syndromic surveillance is a location-sensitive activity. Being a public health effort, agents of syndromic surveillance will usually be acting in the interest of a nation or local government. Consequently, any efforts to employ Twitter for syndromic surveillance will have to take the geographical location of

¹<https://www.urbandictionary.com>

²<https://www.mongodb.com>

	Counts
Tweets	10,702,063
URLs	2,225,155
Hashtags	177,506
Emojis	3,103,598
Number of unique users	5,861,247
Number of tweets per user	4.1

Table 4.1: Information on the data corpus collected before cleaning

Tweets into consideration. Because this project is concerned with means of using Twitter data for syndromic surveillance systems in England, we would like to exclude tweets originating from outside England. While doing so will yield a more useful signal, it is a non-trivial problem and an active area of research [4]. Although Twitter users have the option to disclose their city-level location, fewer than 14% of users do so [74], and up to 30% may give false or fictitious locations [192]. Less than 0.5% of users turn on the location function, which would give accurate GPS coordinate information, owing to concerns over privacy.

The *time_zone*, *coordinates* and *place* attributes, which we collected, can

Data Collection Period	Percentage of Tweets Containing attributes		
	Coordinates	Timezone	Place
September 23, 2015 - November 30, 2015	0.30%	57.90%	2.17%
June 15, 2016 - August 30, 2016	0.29%	61.12%	2.10%
January 27, 2017 - July 31, 2017	0.21%	59.21%	1.61%

Table 4.2: Availability of geolocation attribute in collected Twitter Data

help in the geolocation of a tweet but are not always present. The *time_zone* attribute can be optionally set by a Twitter user as part of their Twitter profile and may not be accurate, or may represent the user's home location but not that where a specific tweet originated. The *place* attribute is also optional for the user to set with the same caveats. The *coordinates* attribute is available when a user chooses to publish their location at the time of tweeting from a mobile device. The *coordinates* attribute is the most reliable, but only present in a small percentage of Tweets, as shown in table 4.2. Therefore, we employ all three geolocation attributes, filtering out tweets that do not have a UK timezone, a place in the UK or coordinates in the UK.

4.4 Data Cleaning and Preprocessing

After the data was collected, we examined its suitability for our purpose. In doing so, we noticed the following data quality problems, which we solved, developing suitable pre-processing algorithms where necessary.

4.4.1 Retweets

A Retweet (sometimes abbreviated to RT) is a re-post of a Tweet. The Retweet feature allows users to quickly share an existing post - which could have been made by them or some other user - while also attributing creation of the post to the original poster [76]. If some user finds a Tweet relatable, they may decide to retweet it. This could lead to duplication in our dataset and could result in the raising of false signals. Retweets are always of two forms: either they contain the original post with the username of the original poster in quotes; or they just contain the original post along with the username of the original poster with the word “RT” preceding it or following it. Tweets in our dataset which satisfied either of these criteria were removed.

4.4.2 Duplicate Tweets

Even when not immediately identifiable as Retweets, Tweets containing exactly the same text with maybe different URLs appended may be duplicates and may give rise to amplified signals. Even after removing Retweets, we noticed a significant number of duplicate Tweets of this nature, many associated with news items or blogs. Those were dealt with by removing Tweets that contained exactly the same text, once URLs were discounted. However, it is worth noting that it is possible for different people to express the same sentiment with the same or very similar words, as twitter encourages very short communication (e.g. “asthma bad” or “can’t breathe”). We also reasoned that a user expressing a condition should only be counted once per day for the purposes of syndromic surveillance, so we removed multiple Tweets for the same user on a given day.

4.4.3 URLs

Some Tweets contain web links to other pages. Usually these links point to pages which explain the content of the Tweet in more detail. However,

Tweets containing URLs often constitute external reports (such as news articles and blog posts), as opposed to individual reports (ie. an individual expressing concern or opinion). Because of this, we removed URLs and replaced them with the token “<URL>”. This not only allowed us to avoid introducing noise to our system, but was also helped us recognize Tweets that originate from individual user accounts as opposed to news and spam accounts.

4.4.4 Spambots and Articles

A “bot” is the term used for when a computer program interacts with web services that are intended for human use. It is possible to create a Twitter account and then through the use of the Twitter API automatically post tweets, follow other users and even send direct messages to other users. Tweets made by such accounts are not genuine individual sentiments and so are not of any relevance to our endeavour. News accounts and web blog accounts which usually post links to news and blog articles fall into a similar category and are not relevant in the context of our investigation. Tweets made from such accounts were removed from the dataset.

To recognise those, we looked for Tweets containing a URL, and then we check if the user had a very skewed following/follower ratio and a relatively high number of Tweets as those characteristics may be associated with spam accounts [276]. We trained and employed a K-Nearest-Neighbour (KNN) [7] classifier to automatically determine which Tweets were not posted by individuals.

	Counts
Tweets	127,145
URLs	147,102
Hashtags	23,189
Emojis	36,872
Number of unique users	115,583
Number of tweets per user	5.3

Table 4.3: Information on the data corpus collected after cleaning

4.4.5 Labelling

3,500 Tweets from the first data collection period were randomly sampled and labelled as “relevant” or “not relevant”. A Tweet was labelled as relevant if it declared or hinted at an individual experiencing symptoms pertain-

ing to the syndrome of choice - *asthma/difficulty breathing*. The labelling was done by three volunteers from the computer science department of the University of East Anglia. A first person initially labelled the Tweets. This took approximately 1 hour per 1,000 Tweets. A second volunteer checked the labels, and flagged up any Tweets with labels that they did not agree with. These flagged Tweets were then sent to a third volunteer who made the decision on which label to use. 23% of the labelled Tweets were labelled as “relevant” while 77% were labelled as “irrelevant”. A second set of 2,000 Tweets, selected at random, were later labelled following the same procedure from the third data collection period. 32% of these Tweets were labelled as relevant and 68% were labelled as irrelevant. For the sake of clarity, we refer to the first labelled dataset as *Dataset A* and the second labelled dataset as *Dataset B*. Together, these two datasets make up our total collection of labelled Tweets, which we refer to as *Dataset L*. The Inter-Rater Agreement was computed using Fleiss’ Kappa [80] which is given by the following equation:

$$\kappa = \frac{\bar{P} - P_e}{1 - P_e}$$

where \bar{P} is the mean agreement between raters and P_e is the probability of agreement by chance calculated from the observed data using the probabilities of each rater randomly labelling a tweet as relevant or irrelevant. The Fleiss’ kappa was chosen over other kappas due to the fact that it is intended to be used when assessing the agreement between three or more raters which is appropriate for our scenario. A value of 1 suggests complete agreement while a value of 0 suggests total disagreement. We obtained a value of **0.906** for κ .

4.5 Feature Extraction

A goal of this thesis is to develop ways of dealing with Twitter data using semi-supervised means. Using semi-supervised methods enable us to use a small amount of labelled data, thereby reducing the initial labelling effort required to build a classifier. Relatively little research has gone into semi-supervised learning for Twitter mining as was evidenced by our scoping review in chapter 3. This is most likely owing to the fact that it is difficult to develop useful machine learning systems with limited data. Nonetheless, due to its merit, we set our sights on developing semi-supervised systems for syndromic surveillance using Twitter.

Although it is possible to use any sequence of letters or language tokens to represent text, words are often identified and used in text mining. Word n -grams have been used successfully in language modelling and speech recognition [124, 293, 203]. Words are identified and extracted after a process of tokenisation and can then be used to represent a document by their pres-

ence or absence without trying to retain any information on the ordering of words or their relationship to one another. That approach is called “bag of words” and despite its relative simplicity can work well in many text mining scenarios [122]. However, some authors [18, 145, 287] have argued that more complex features will dramatically decrease the feature space while leading to better classification performance. More recently, deep learning has been utilized to extract features from raw text data. Word embeddings (sometimes referred to as word vectors) are a powerful distributed representation of text learned using neural networks. Word embeddings are often used to encode semantic information of texts in dense low-dimensional vectors, and have been shown to perform well in similarity-based tasks [117].

Classification of Tweets may be challenging as they are texts which are usually very short in length. Furthermore, in our scenario, the classes may share common vocabularies. That is, both relevant and irrelevant Tweets contain the same words. Twitter has specific language and styles of communication that people use. In particular, we found that *emojis* and *emoticons* are promising additional tokens that we could exploit in classification:

- An emoticon is a pictorial representation of a facial expression using punctuation marks, numbers and letters, usually written to express a person’s feelings or mood. :-) is an example of an emoticon.
- Emojis on the other hand are miniature graphics of various objects and concepts including facial expressions. 😊 is an example of an emoji.

Emoticons have previously been used successfully as features for Tweet classification performance in sentiment analysis, as well as syndromic surveillance [145]. In a pragmatic sense, emojis can be used for the same purposes as emoticons. However, emojis have seen a recent surge in popularity, presumably due to the fact that emojis provide colourful graphical representations, as well as a richer selection of symbols [161]. In fact, as table 4.1 shows, we observed a large number of emojis in our corpus. A further advantage of emojis is that while emoticon use can differ around the world, emoji features are less variant. Take for example the crying emoticon. In Western countries, it is usually represented by :(or :-(. In Asian countries, “kaomoji”, which refers to emoticons depicting faces and made from a combination of Western punctuation characters and CJK (Chinese-Japanese-Korean) characters, are more popular than regular emoticons [22]. An example of a popular kaomoji is “ㄟ(ㄋ)ㄟ”. Now, using the earlier example of the crying face, we could now also expect to see (T_T) for the same concept. Emojis on the other hand are a predefined set of unicode characters. Even though they may be rendered differently on different devices, the underlying mapping between a concept and an emoji remains the same. In this sense, emojis may transcend language barriers. In light of these observations, we extended existing emoticon feature techniques with

the inclusion of emojis.

We believe that emoticons and emojis can help with assessing the tone of a tweet. Tweets which we are interested in, will most likely have a negative tone as they reflect people expressing that they are unwell or suffer some symptoms. This means that they may contain one or more sadness, anger or tiredness-related emojis/emoticons, for example. On the other hand, the presence of emojis/emoticons denoting happiness and laughter in a Tweet may be an indication that it is not relevant to our context of syndromic surveillance. In this section, we explore different avenues for feature extraction and representations for our Tweets, including but not limited to complex features derived from ideas such as emojis.

4.5.1 Word Classes

Word classes are labels that Lamb et al. [145] found useful in their efforts to analyse Tweets and categorise them as related to infection or awareness. The idea behind word classes is that many words can behave similarly with regard to a class label. A list of words is created for different categories such as “*possessive words*” or “*infection words*”. Word classes function similarly to bag of word features in that the presence of a word from a distinct class in a Tweet triggers a count based feature. We manually curated a list of words and classes which are shown in table 4.4. As we applied lemmatisation, we did not include multiple inflections of the words in our word classes.

Word Class	Member Words
Infection	sick, down, ill, infect, caught, recover
Possession	have, contain, contaminated, my
Concern	awful, worried, scared, afraid, terrified, fear, sad, unhappy, feel
Humour	laugh, ha, haha, hahaha, lol, lmao, rofl, funny, hilarious, amused
Symptomatic	runny nose, cough, spray, shots, wheezing, mucus, cold

Table 4.4: Our list of word classes with their member words

4.5.2 Positive and negative word counts

We constructed two dictionaries of positive and negative words respectively. These dictionaries are included in the appendix. For every tweet, the number of positive words and negative words contained within it is computed. Our manually curated dictionaries are used as a reference point for which words are positive or negative. Words which do not appear in either of our dictionaries are not counted. This is because our dictionaries are concerned specifically with positive and negative words which are likely to appear in the context of health. In essence, this feature produces two figures for every tweet - a positive count and negative count. Our hypothesis is that tweets which contain more negative words than positive words are likely to be relevant in the sense that they are an individual reporting symptoms or expressing concern over a syndrome. It is then the duty of our learning algorithm to learn a matching between ratios of positive to negative counts to tweet relevance.

4.5.3 Indicates Asthma Possession

This feature tests for the presence of the word “asthma” in close proximity together with a personal pronoun. In particular we check for close proximity with “i’m”, “im”, “my”, “i”, “am” and “me”. The aim of this feature is to determine when a tweet has a user reporting concern over their condition and distinguish this from a tweet where a tweet just happens to mention asthma. For some perspective, only 44% of the Tweets in the dataset contain the word “asthma” and 35% of these Tweets are relevant. When applied to the dataset, the feature had a value of “True” in 8% of the dataset, and of this 8%, 53% were relevant. As such, this feature shows that it is better to check that a tweet contains “asthma” used in a particular way than to just check whether a tweet contains “asthma”, and is one we decided to investigate.

4.5.4 Contains “Asthma-Verb” Conjugate:

This is a very specific feature to our syndrome. A verb conjugate is a form of a verb derived from its base-form according to the rules of grammar, due to a change in person, tense, number or other grammatical categories [202]. *Contains “Asthma-Verb” Conjugate* is a syntactic binary feature which is concerned with whether or not there is a verb form appearing with the word asthma (or its symptomatic words) as its object. For example, the Tweet “I can’t believe I’m only just recovering from my asthma attack” contains the word *asthma* used as the object of the verb *recover*, while the Tweet “People with asthma shouldn’t come to school” sees it being used as part of the subject of the verb. The WordNet³ interface of NLTK (Natural

³<https://wordnet.princeton.edu>

Language ToolKit) was used to perform Part of Speech (POS) tagging in order to extract these type of features. NLTK is a platform for building Python programs to work with human language data. As the same words may appear in both classes, it can sometimes be problematic to rely solely on features based on the presence or absence of a word. We hypothesize that using syntactical features may help us alleviate such issues.

4.5.5 Denotes laughter:

This is a simple binary feature which measures the presence of an emoji and/or emoticon that might suggest laughter or positivity. We manually curated and saved a list of positive emojis/emoticons for this. The usefulness of this feature was augmented by also checking for the presence of a small list of more established and popular internet slang for laughter or humour such as “lol” or “lmao” which stand for “Laughing Out Loud” and “Laughing My Ass Off” respectively.

4.5.6 Negative emojis/emoticons:

This is similar to the *Denotes Laughter* feature but this time looking at the presence of an emoji or emoticon that can be associated with an illness or the symptoms that it may bring, i.e. negative emotions. We decided to include these features because we discovered the ubiquity of emojis on Twitter and wanted to investigate their potential. Table 4.5 shows this feature’s distribution over the data. We find that this feature may be the most discriminative of the two emoji-based features. Of the instances with a positive value, a high percentage belong to the “relevant” class and of the instances with a negative value, a high percentage belong to the “not relevant” class.

For each tweet, we can append all of the above features together to form one feature vector. Each Tweet T_i is therefore represented by an f dimensional vector, where f is a sum of the number of terms, n , in the constructed vocabulary, and the dimensionalities of our custom features C (*Word Classes*, *Positive and Negative Word Counts*, *Contains Asthma-Verb Conjugate*, *Indicates Asthma Possession*, *Denotes Laughter* and *Negative Emojis/Emoticons*). This gives us

$$T_i = \{t_{i1}, t_{i2}, \dots, t_{in}\} \cup \{C_i^1\} \cup \{C_i^2\} \cup \{C_i^3\} \cup \{C_i^4\} \cup \{C_i^5\}$$

where t_{ij} represents the weight of the j -th vocabulary term in the i -th Tweet and C_i^k represents the value of the k -th custom feature in the i -th Tweet.

Feature	Value Distribution		Class Distribution	
			Relevant	Not Relevant
<i>Contains Asthma-Verb Conjugate</i>	TRUE	20.9%	45.6%	54.4%
	FALSE	79.1%	18.9%	81.1%
<i>Indicates Asthma Possession</i>	TRUE	8.3%	53.6%	46.4%
	FALSE	91.7%	20.1%	79.9%
<i>Denotes Laughter</i>	TRUE	3.9%	31.8%	68.2%
	FALSE	96.3%	24.2%	75.8%
<i>Negative Emojis/Emoticons</i>	TRUE	5.5%	74.8%	25.2%
	FALSE	94.5%	21.6%	78.4%

Table 4.5: Distribution of constructed features and classes across the dataset

4.5.7 Text Embeddings

Text embeddings are models which learn a mapping from a set of words and/or phrases to numerical vectors. Word embeddings map words to dense distributed low-dimensional vectors. While training a neural network for some task, and estimating its weights and biases, we can also implicitly learn/estimate embeddings for words in a vocabulary. In this embedding space, similar words are close to each other. For example, the vector for ‘dog’ should be close to the vector for ‘puppy’. This representation will allow us to perform interesting vector operations such as **kitten** – **cat** + **dog** \approx **puppy**. This means that semantic information can be inferred from the vectors as opposed to merely syntactical or count-based information. In addition, such vectors are of a fixed size that is independent of the vocabulary size. A word vector can have a length of 200 or some other arbitrary size selected based on trial or error or some other heuristic. This reduces dimensionality and saves significant computational and memory overheads.

There are two algorithms which have seen widespread use for computing word vectors. The first is word2vec [177] which has two different architectures namely *Skipgram* and *Continuous Bag-Of-Words (CBOW)*. The second is Global Vectors for Word Representation (GloVe) [205]. We built our word vectors on a set of 5 million unlabelled Tweets collected without any keyword restrictions. Our implementations were tested on similarity by using a random sample of words, converting them to word vector space and determining the words most similar to each of them, as the words whose vectors were closest to the vectors of the query words. Table 4.6 shows the

Word	Similar Words
china	tourists, demonstration, descent, 247, germany, octobers, hgv, round
kids	lowest, action, syrup, w, birth, tapped, 43, till
controversy	breathenncos, againlol, #hypocrite, weightlifter, maseratis, #wemissboris, #americasnexttopmodel, defended
fit	mad, sext, 2hrs, blurred, ellen, helped, impotent, blocked
obese	#londonsair, 🙄, cops, #euref, included, choking, scientifically, suffer

Table 4.6: A random sample of words and their 8 most similar words as computed from a Twitter dataset using Skipgram embeddings.

results of our Skipgram word vector model. For some of the words (e.g. China and demonstration or China and hgv or hypersonic glide vehicle) the connection is somewhat obvious, whereas for others, it is more opaque. We also see that this approach can establish connections between words and hashtags or emojis giving more possibilities for expanding vocabularies.

While we have word embeddings, the data we are dealing with is largely in the form of Tweets, which are collections of words. This means that we still need to combine the word vectors within a Tweet in a meaningful way, which preserves the useful semantic relationships between constituent words such that we obtain a powerful understanding of the Tweet as a whole. One way of achieving this is by computing the mean of the word vectors in a text, and using that mean vector to represent the text as a whole. However, in this way, we lose some of the positional information of the text. An alternative is to concatenate the vectors, but this does not represent the complex relationships between the different words particularly well. A more involved solution would be to learn vectors for entire documents. From an NLP point of view, we can view a Tweet as a *document*. For the construction of vector representations for documents, there are models which are extensions to the word embedding models that we can adopt. One such model is *paragraph2vec* [147] which is an extension of the word2vec model. While word2vec has the Skipgram and CBOW variants, paragraph2vec ex-

tends them to the *Distributed Memory Paragraph Vectors (PV-DM)* and the *Distributed Bag of Words Paragraph Vector (PV-DBOW)*. We implemented both variants of the paragraph2vec model, building them from the same Twitter dataset that the word2vec models were built. We tested our paragraph2vec models by way of similarity as before. Table 4.7 shows the results obtained from our PV-DM paragraph2vec model. Again, we can observe that some meaning and semantic similarities are being captured by this approach. We built our text embeddings from our around 5 million Tweets.

Tweet	Similar Tweets
do you know an elderly person with a bad cough trouble breathing a cold or sore throat get advice from nhs direct before it gets worse	might go to casualty and see if i can get an inhaler worth a try anyway
	<usermention> i know a few with asthma and peanut allergies
usermention but what is that i cant even breathe	i cant breathe what even
	usermention hannah im wheezing i dont even need the translation

Table 4.7: A random sample of Tweets and their 2 most similar Tweets as computed using PV-DM embeddings.

We experimented with different hyperparameters when building our embeddings and report the best performing combination. Our word vectors were built using a neural network with the following hyperparameters: We used batch sizes of 128, context windows 1 word wide. For learning weights we used a cross entropy loss function with an Adagrad optimizer[270] with a learning rate of 1.0 and ran the optimization for 100 epochs. As for our document vectors, they were built using the popular gensim python library. Similarly, we used a batch size of 128 and a window size of 1. The document embedding models were trained for 100 epochs.

4.6 Summary

In this chapter, we described the Twitter API and its workings. We explained how we made use of it to collect Tweets in real-time for analysis and syndromic surveillance. We described the data contained within a Tweet and performed some content analysis on our collected Tweets. We found that emoji use was ubiquitous among Tweets. We developed and implemented a number of compound features from the data. One of these features took a novel approach, making use of emojis in its computation. We also implemented popular text embedding features using our large volumes of data. Following this, we look towards employing our extracted processed data in its different feature representations.

Chapter 5

Experimental Methodology: Semi-supervised Classification for Relevance Filtering

5.1 Introduction

In order to assess the utility of Twitter for syndromic surveillance, we must be able to efficiently extract a relevant signal from it. To achieve this, we must be able to effectively identify and extract Tweets expressing discomfort and/or concern related to a syndrome of interest, and reflecting current events. Simply relying on keyword-based data collection, many of the tweets collected are not be relevant because they represent chatter, or talk of awareness instead of suffering a particular condition. Using the keyword filtering capabilities of the Twitter API during data collection as described in chapter 4, we get rid of a large portion of irrelevant Tweets. However, most of the Tweets we collect may mention keywords such as “asthma”, “air pollution” or “wheeze”, but may not necessarily be relevant in that they do not represent a user expressing discomfort. For some context, examples of Tweets containing the keyword “asthma” include “*oh I used to have asthma but I managed to control it with will power*” or “*Does your asthma get worse when you exercise?*”. However, we do not consider these Tweets relevant. On the other hand, Tweets such as “*why is my asthma so bad today?*” express a person currently affected and we would like to consider such a Tweet as relevant. In light of this, we set out to automatically identify relevant tweets to collect a strong and reliable signal.

We look at the problem of relevance filtering as a text classification task. In particular, we focus on semi-supervised techniques. We are able to collect

large amounts of Twitter data with relative ease. In this work, we have hundreds of millions of Tweets. We are only able to label a minuscule percentage of this data. Semi-supervised learning will not only reduce the labelling burden, but also allow us to capitalize on the vast amounts of data available to us. In this chapter, we discuss the semi-supervised classification algorithms that we developed and applied to our Twitter data. We also investigated the capacity of deep learning as a solution to the relevance filtering problem. We apply the feature representations put forward in chapter 4 in order to prepare the data for manipulation by the classification algorithms.

5.2 Iterative Labelling Algorithms

Iterative labelling algorithms refer to a family of semi-supervised techniques which select and label unlabeled data in an iterative process [94]. In particular, we make use of **self-training**, and describe our own incarnation of the algorithm. We then build on our self-training approach, extending it in order to make it more robust through **co-training**.

5.2.1 Self-Training

Self-training is an iterative labelling algorithm that is closely related to, and is essentially an extension of the Expectation-Maximization (EM) algorithm put forward by Dempster et al. [70]. It is a sort of *meta-algorithm* which uses a data set S of labelled instances L , unlabelled instances U , and a supervised learning algorithm A with

$$S = \{L \cup U\}$$

Self-training aims to derive a function f which provides a mapping from S to a new dataset S' :

$$f(S, A) = S' \leftrightarrow \{L' \cup U' \} \mid |U'| \leq |U|, |L'| \geq |L|$$

Such an algorithm can be defined simplistically as an iterative execution of three functions: *Choose-Label-Set*(U, L, A) selects and returns a new set, R , of unlabelled examples to be labelled; *Assign-Labels*(R, A) generates labels for the instances selected by *Choose-Label-Set*(U, L, A); *Stopping-Condition*(S, S') dictates when the algorithm should stop iterating. For our choice of supervised learning base-algorithm, we selected the Multilayer Perceptron (MLP) classifier after experimenting with different supervised models and finding it to perform best. We used the trained MLP classifier's predictions to label previously unlabelled instances in the *Assign-*

Algorithm 1 Iterative labelling Algorithm

```

1: function TRAINCLASSIFIER( $A, L$ )
2:   return  $A(L)$ 
3: end function
4: function ITERATIVELABELLING( $U, L, A$ )
5:   repeat
6:      $A \leftarrow \text{TrainClassifier}(A, L)$ 
7:      $R \leftarrow \text{Choose-Label-Set}(U, L, A)$ 
8:      $R' \leftarrow \text{Assign-Labels}(R, A)$ 
9:      $U \leftarrow \text{Replace-Instances}(U, R')$ 
10:     $S' \leftarrow R' \cup L \cup U$ 
11:   until  $\text{Stopping-Condition}(S, S') = \text{True}$ 
12: end function

```

Labels function. We set our stopping condition such that the iteration stops either when all the unlabelled data is exhausted or when there begins to be a continued deterioration in performance with the labelling of more data. Along with the class of an applied instance, we also compute the model's confidence in its classification. Our algorithm, inspired by Truncated Expectation-Maximization (EM) [148], then grows the labelled set, L , based on the confidence of our model's classification. When an instance from R is classified, if the confidence of the classification is greater than some predetermined threshold θ , the instance is labelled. With this in mind, our algorithm falls within the *confidence-based* category of iterative labelling algorithms because it selects instances for which the trained classifier has a high confidence in its predictions.

Confidence-based iterative labelling algorithms can tend toward excessively conservative updates to the hypothesis, since training on high-confidence examples that the current hypothesis already agrees with will have relatively little effect [70]. Furthermore, it has been proven that in certain situations, many semi-supervised learning algorithms can significantly degrade the performance relative to strictly supervised learning [50, 211]. Because of this, we make extra considerations around our self-training algorithm, extending it using *co-training*.

5.2.2 Co-Training

To address the issues associated with self-training, we take some ideas from *co-training* [24]. Co-training requires different views of the data so that multiple classifiers can be maintained for the purpose of labelling new instances. Recall that each Tweet can be represented as a feature vector T_i containing various features. We now distinguish two representations. The first is a concatenation of our *n-grams*, *Word Classes*, *Denotes Laughter* and *Negative Emojis/Emoticons* features. We represent this feature space

as X_1 . The second kind of feature vector is a concatenation of our n -grams, *Positive and Negative Word Counts*, *Denotes Laughter* and *Negative Emojis/Emoticons* features. We represent this feature space as X_2 . We can think of X_1 as the **taxonomical** feature space as is characterised by its inclusion of the *Word Classes* feature while X_2 can be the **sentimental** feature space and this is characterised by its inclusion of the *Positive and Negative Word Counts* feature. As such, X_1 and X_2 offer different, though overlapping, views of the dataset. Each tweet is then represented as a feature vector from each of these spaces.

With this setup, we now maintain two separate classifiers trained on different views of the data. During the iterative labelling process, we only label instances for which at least one of the classifiers has a high confidence in its prediction, and take the result of that classification as the label. Similar to self-training, at the end of each iteration, the newly labelled data is incorporated into each of the classifiers to update their hypotheses. On completion of the iterative labelling process, the prior training examples for both classifiers, as well as the newly labelled examples are joined together and used to train a new classifier using all the features which will then be applied in practice. The benefit of co-training is that the instances labelled by one classifier are also presented to another classifier to update the hypothesis on a complementary view. Thus, the examples, as represented in each view, receive at least some of their labels from a source other than the classifier that will be updated with them.

5.3 Generative Classification Network

We investigate the use of deep neural networks for our task. According to the *universal approximation theorem*, deep neural networks are universal function approximators. The universal approximation theorem states that a feed-forward network with a single hidden layer containing a finite number of neurons (i.e. a multilayer perceptron), can approximate continuous functions on compact subsets of \mathbb{R}^n , under mild assumptions on the activation function. It has been shown that it is not the specific choice of the activation function, but rather the multilayer feedforward architecture itself which gives neural networks the potential of being universal approximators [105]. With all this in mind, we introduce a different approach to text classification and relevance filtering obtained from combining machine learning techniques - the **Generative Classification Network (GCN)**. The GCN is a semi-supervised text classification algorithm that makes use of neural language models built on both labelled and unlabelled data to perform classifications.

Given a sequence of words, neural language models predict the probability of a word occurring next given the previous words. They also allow us

to measure how likely a sentence is. However, these models can also be used in a *generative* context. That is, they can be used to generate new texts by sampling from the output probabilities. GCNs take advantage of this, and classify texts in a semi-supervised manner.

5.3.1 Architecture

The Generative Classification Network relies on two or more generative neural models. Recurrent Neural Networks (RNNs) are a category of neural networks that incorporate sequential information, and are well suited to language modelling [170]. While in a traditional neural network, inputs are independent, in RNNs each node depends on the output of its preceding node. This is particularly useful for sequential data, such as text, where each word depends on the previous one. While in theory, RNNs can make use of information in arbitrarily long lengths of text, practically speaking, they are limited to looking back only a few steps due to the vanishing gradient problem which occurs during the back-propagation algorithm. When tuning the parameters of the network, due to long sequences of matrix multiplications, gradient values shrink fast and gradient contributions from earlier neurons become zero. As a result of this, information from earlier inputs (words in the text) do not contribute to the overall algorithm. Long Short Term Memory (LSTM) networks are a flavour of the RNN architecture which make use of a gating mechanism to combat the vanishing gradient problem. In our implementation, we make use of LSTM RNNs for our generative neural language models. A regular neural network would simply consist of a single layer with an activation function which is related to the output as below, where w_i represent the weights and a_i the inputs for all L layers:

$$\phi = \sum_{i=1}^L w_i \cdot a_i \quad (5.3.1)$$

$$y = \tanh(\phi) \quad (5.3.2)$$

The LSTM model adds some complexity to the regular neural network architecture. The LSTMs we used for our generative model contained only one LSTM layer. The network has an input layer x , hidden layer h , LSTM cell state c and output layer y . Input to the network at timestep t is $x(t)$, output is denoted as $y(t)$, hidden layer state is $h(t)$ and LSTM cell state is $c(t)$. The LSTM cell state is controlled by the gating mechanism as highlighted above briefly. Each cell consists of the following gates which interact with each other to dictate the overall cell state:

- input gate (i)
- forget gate (f)

- write gate (g)
- output gate (o)

Each of these gates has its own weights and biases and is a function of the previous timestep's hidden state $h(t-1)$. The hidden state of a layer can then be computed as a function of the cell state as shown below:

$$c(t) = f(t) \cdot c(t-1) + i(t) \cdot g(t) \quad (5.3.3)$$

$$h(t) = o(t) \cdot \tanh(c(t)) \quad (5.3.4)$$

For the sake of brevity and simplicity of our equations, let us assume that there is only one hidden layer l so that we do not have to specify different equations for the different edge cases that would come with multiple layers, such as when execution is in the first layer and has no previous layer or when it is in a middle layer or the final layer. In the real world scenario, this is not the case as each hidden layer state is influenced by the hidden state in the previous timestep as well as the state of the previous hidden layer. To adapt this, one may simply add the product of the weights and input of the previous layer to each activation function. The activation functions for the gates are computed as:

$$f(t) = \text{sigmoid}(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \quad (5.3.5)$$

$$g(t) = \tanh(W_{xg} \cdot x_t + W_{hg} \cdot h_{t-1} + b_g) \quad (5.3.6)$$

$$i(t) = \text{sigmoid}(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \quad (5.3.7)$$

$$o(t) = \text{sigmoid}(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o) \quad (5.3.8)$$

where W_{pq} are the weights that map p to q and b_p refers to the bias vector of p . For example, if we look at equation 5.3.5, W_{xf} refers to the weights going from input x to the forget gate f and so on while b_f refers to the bias of the forget gate f .

Our language models were trained on the word level using GloVe vectors to represent each word. Our models predicted the next single word for an input of three words. Our models were 512 layers deep with a learning rate of 0.001. The model for the irrelevant class had more epochs and was trained over 40,000 iterations while that of the relevant class was trained over 17,500 iterations. We did this because there were a lot more irrelevant tweets than relevant tweets. This meant that the irrelevant model had more data to be built on than the relevant model, so couldn't be trained in the same way in order to avoid overfitting one class.

5.3.2 Algorithm

The generative classification network employs neural language models for the purpose of text generation. For each class $c \in C$, a language model M_c is constructed from the unlabelled text and texts pertaining to that class. Recall that a language model predicts the probability of a piece of text (essentially a sequence of words) occurring. Given a test text with an unknown class, each of these language models can then be applied to this text in order to predict how probable it is for that text to occur within the context of the language model which is concerned with a particular class. The class c of the model M_c which yields the highest probability should be assigned to the test text. Note that in our problem scenario, a ‘text’ refers to a Tweet, but this algorithm may be applied to other problems so we generalise our descriptions by using ‘text’, which simply refers to an arbitrarily long sequence of words.

More formally, we can define the algorithm given the set of labelled texts L and unlabelled texts U and the set of classes C with k possible classes.

$$C = \{c_1, c_2 \dots c_k\} \quad (5.3.9)$$

L is partitioned into k disjoint sets with each disjoint set containing only texts labelled to a particular class.

$$L = L_{c_1} \sqcup L_{c_2} \sqcup \dots \sqcup L_{c_k} \quad (5.3.10)$$

For each class c , language models M_{c_i} are built using labeled texts with that class’ label and the unlabelled texts.

$$\forall L_{c_i i \in \{1, \dots, k\}}, M_{c_i} = LSTM(U, L_{c_i}) \quad (5.3.11)$$

The probability of a given text x belonging to a class c_i is equal to the probability with which a language model M_{c_i} predicts the text as likely. We can then assign a class to x as shown below:

$$P(c_i) = P(x|c_i) = M_{c_i}(x) \quad (5.3.12)$$

$$c = \operatorname{argmax}_{i \in \{1, \dots, k\}} P(c_i) \quad (5.3.13)$$

5.4 Attentive Bi-directional Recurrent Neural Network

Much of the difficulty in classifying Tweets comes from the fact that they are usually very short texts. Most popular text classification algorithms expect rich informative texts as input. For this reason, we propose an attention-based approach to short text classification, which we have created for the practical application of Twitter mining for syndromic surveillance. We reason that with the lack of surplus of (informative) words, it is useful for our models to know which words or regions of text are important, particularly with regards to informing the relevance of a Tweet.

Traditional text classification approaches assume that independent keywords or phrases are important to the text category and extract vector features representing those keywords or phrases using statistical methods [234]. These methods generally yield successful results but the assumption is an oversimplification that brings some shortcomings. While independent keywords and phrases are important, there are other linking words which also give meaning to a text. The way words relate can provide context and disambiguation and without this, we potentially lose some information. Recently, deep-learning-based methods have seen a lot of success for text classification. This is largely due to the fact that such methods can automatically and effectively learn underlying features and interrelationships in data. But while deep learning models have seen widespread success, they treat all the words in a text as blocks of input without giving any words or phrases special treatment. We would like to leverage the advantages of both the classical text categorization approaches, which employ keywords, and the modern deep learning approaches, which learn underlying relationships, for text classification.

We experiment with a bi-directional Recurrent Neural Network architecture with an attention layer (termed ABRNN) which allows the network to weigh words in a Tweet differently based on their perceived importance. We further distinguish between two variants of our ABRNN based on the Long Short Term Memory and Gated Recurrent Unit architectures respectively, termed the ABLSTM and ABGRU. We combine the self-learning and intrinsic pattern recognition capabilities of deep learning, with the use of keywords in classification that is typically employed by traditional classification methods.

In this section, we describe the proposed attention-based RNN. The model can be broken down into four parts:

1. **Word Embedding:** This step vectorises the Tweet. It involves mapping each word in the Tweet to a fixed-dimension word embedding. In our work, we make use of GloVe embeddings which we build from

a large unlabelled corpus of Tweets.

2. **RNN**: Takes the output of the previous step as input. The RNN learns high level features from the given input.
3. **Attention Layer**: Produces a weight vector which it uses in conjunction with the output states of the RNN to form a new Tweet representation.
4. **Classification**: The attention-powered vector representation of the Tweet is fed into a classifier to obtain a prediction

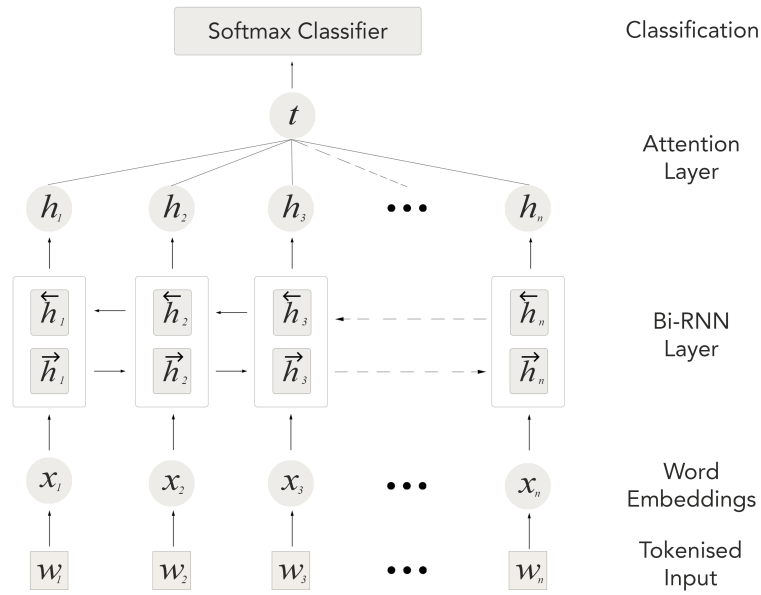


Figure 5.4.1: Attention-based RNN model

Figure 5.4.1 illustrates the architecture of our attentive RNN model. Each component of the model will subsequently be explored in more detail below.

5.4.1 Word Embeddings

Word embeddings as described in chapter 4, (sometimes referred to as word vectors), are a powerful distributed representation of text learned using neural networks that have been shown to perform well in similarity tasks [118]. They encode semantic information of words in dense low-dimensional vectors. After we build a word embedding model, an embedding matrix X of size $|V| \times d$ is produced where V is the set of all the words in our vocabulary and d is the dimension of each word embedding. Given a Tweet T consisting of n words, $T = \{w_1, w_2, \dots, w_n\}$, each word w_i is converted to a real-valued vector x_i by performing a lookup from the embedding matrix

X. For this work, we built GloVe embeddings [206] from a set of 5 million unlabelled Tweets.

5.4.2 Bi-directional Recurrent Neural Network

Similarly to the generative classification model (GCN) described in section 5.3.1, this model makes use of RNNs. Recall the vanishing gradient problem which plagues RNNs and can be solved by a number of flavours of the RNN architecture which make use of gated mechanisms. In section 5.3.1, we make use of LSTMs as a workaround for the vanishing gradient problem. In our attentive bi-directional RNN model, we again make use of LSTMs, but also employ Gated Recurrent Unit (GRU) RNNs. The GRU is another solution for the short-term memory problem that simple RNNs possess, where they cannot properly update and learn weights for earlier inputs in a sequence. LSTMs and GRUs are very similar, the main difference is that GRUs have less parameters than LSTMs. As such, GRUs are faster and have been observed to exhibit better performance on some smaller datasets [45]. However, LSTMs have been shown to be better at learning in general [272]. We have already described LSTMs in detail in section 5.3.1. We will now describe the GRU flavour of the RNN.

Gated Recurrent Unit (GRU) Again, for the sake of brevity and simplicity of our equations, let us assume that there is only one hidden layer l . The GRU cell state is controlled by a gating mechanism, similar to the LSTM. Each cell consists of the following gates which interact with each other to dictate the overall cell state:

- update gate (z)
- reset gate (r)

The gates can be formalised as follows:

$$z(t) = \text{sigmoid}(W_{xz} \cdot x_t + W_z \cdot h_{t-1} + b_z) \quad (5.4.1)$$

$$r(t) = \text{sigmoid}(W_{xr} \cdot x_t + W_r \cdot h_{t-1} + b_r) \quad (5.4.2)$$

The hidden state of a layer is computed as a function of the input and gates as shown below:

$$h(t) = z(t) \cdot h(t-1) + (1 - z(t-1)) \cdot \tanh(W_x + r(t) \cdot W_h \cdot h(t-1)) \quad (5.4.3)$$

where W_{pq} are the weights that map p to q and b_p refers to the bias vector of p . For example, if we look at equation 5.4.1, W_{xz} refers to the weights going from input x to the update gate z and so on, while b_z refers to the bias of the update gate z and W_z refers to the weights for the update gate

itself.

Bi-directional Networks The above RNNs process sequences in time steps with subsequent time steps taking in information from the hidden state of the previous time steps. This means that they ignore future context. Bi-directional RNNs (Bi-RNNs) extend this by adding a second layer where execution flows in reverse order [231]. Hence, each layer in a Bi-RNN has two sub-layers: one moving forward in time steps and one moving backwards in time steps. To compute the hidden state $h(t)$ of a Bi-RNN layer, we perform an element-wise sum of the hidden states computed from both its sublayers:

$$h(t) = \overrightarrow{h(t)} \oplus \overleftarrow{h(t)} \quad (5.4.4)$$

where $\overrightarrow{h(t)}$ and $\overleftarrow{h(t)}$ are the hidden states of the forward and backward traversals of the bi-directional RNN.

5.4.3 Attention

In this section, we describe the attention mechanism used. The Bi-RNN layer takes in a sequence of vectors for each of the words in an n -worded Tweet $\{x_1, x_2, \dots, x_n\}$, resulting in hidden states $\{h_1, h_2, \dots, h_n\}$ where h_i is a vector derived from equation 5.4.4. That is, the hidden state of the Bi-RNN for the i^{th} word, w_i , is h_i . Let H be the matrix containing these vectors such that $H \in \mathbb{R}^{k \times n}$ where k is the number of neurons in the hidden layer. A Tweet representation t can be derived by taking a weighted sum of the hidden vectors with the attention weight for the relevant words. We represent the attention weights as α , such that α_i represents the attention weight for w_i . α is obtained from trainable parameters and so is adjusted as the optimization algorithm trains the network. We have:

$$M = \tanh(H) \quad (5.4.5)$$

$$\alpha = \text{softmax}(w^T M) \quad (5.4.6)$$

$$t = M\alpha^T \quad (5.4.7)$$

where w is a trainable parameter in the network and w^T is its transpose. w , α and t have the dimensions k , n and k respectively. Finally, the hyperbolic tangent function (\tanh) is applied to t , the Tweet attention vector, in order to squash it between the range $[-1, 1]$ and make it easier to train with the network:

$$t^* = \tanh(t) \quad (5.4.8)$$

5.4.4 Softmax Layer

Once the new attention-based representation for the Tweet has been obtained, it is passed to a softmax classifier to make the class prediction. The softmax layer predicts a class y from a discrete set of m classes Y by calculating the probability with which the observed Tweet belongs to each class, $P(y|T)$, and assigning that Tweet the class for which the highest probability was observed. In more formal terms, we have:

$$P(y|T) = \text{softmax}(W_s t^* + b_s) \quad (5.4.9)$$

$$y = \text{argmax}_y P(y|T) \quad (5.4.10)$$

where W_s represents the softmax classifier network weight and b_s represents its bias term. The loss function we used to train the entire network was the cross entropy loss function [32]:

$$L = -\frac{1}{m} \sum_i^m e_i \log(o_i) \quad (5.4.11)$$

where L estimates the loss between the observed and expected values. e is a one-hot encoded vector of the ground truth for t and o is the probability of each class being the target according to the softmax classifier.

The hyperparameters of the attention networks were selected using grid search. The dimension of our word vectors d was 200. The hidden layer size k was also 200. The learning rate of the optimization algorithm was 0.001. The dropout rate was set to 0.3 and the networks were trained for 50 epochs. The other parameters such as weights and biases were initialised randomly.

5.5 Evaluation and Performance Metrics

5.5.1 Model Evaluation

Accuracy is a statistical measure of how well a binary classifier correctly makes a prediction [176]. Simply put, accuracy is the proportion of correct results among the total number of cases examined and is computed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP},$$

where

TP is the number of true positive cases;

FN is the number of false negative cases;

FP is the number of false positive cases;

TN is the number of true negative cases.

Accuracy can be a misleading measure [263], as it may only be reflecting the prevalence of the majority class. This is known as the accuracy paradox. It means that we could get high values of accuracy by classifying all Tweets as irrelevant (the majority class). This would, however, not improve our signal. Our goal to identify Tweets which might suggest an increase in cases for a particular syndrome, *asthma/difficulty breathing*, for the purpose of syndromic surveillance. The signal for some syndromes can be quite weak as not many cases may occur at a national level and even less may be talked about on Twitter. Because of this, we are very concerned with identifying and keeping instances of the positive class (i.e. relevant Tweets). We would indeed like to reduce the number of irrelevant Tweets, but not at the expense of losing the relevant Tweets. This means that for our classifier, errors are not of equal cost. Relevant Tweets that are classified as irrelevant, also known as False Negative (FN) errors, should have a higher cost and hence be minimised; we can have more tolerance of irrelevant Tweets classified as relevant, also known as False Positive (FP) errors. Those subtleties are well captured by alternative measures of model performance.

Recall is the probability that a relevant Tweet is identified by the model [106], and is defined as:

$$Recall = \frac{TP}{TP + FN},$$

Precision is the probability that a Tweet predicted as relevant is actually relevant [106], and is defined as:

$$Precision = \frac{TP}{TP + FP}.$$

Precision and recall are often trading quantities.

F-measure is a metric that combines precision and recall by taking their harmonic mean [106]. The standard F -measure or balanced F -score is defined as:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

F2 score is a variation of the standard F -measure which weighs recall higher than precision. As such, it may be more appropriate for our purposes. The formula for positive real $\beta = 2$ is defined as:

$$F_\beta = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}}.$$

5.5.2 Feature Evaluation

We assessed the discriminative ability of our features by performing feature ablation experiments [158]. For each feature, we evaluated the performance of a given model when using every feature, and then again after removing this feature. The observed difference is used as a measure of the importance of the feature.

We also performed some analysis on the word (i.e. n -gram) features in order to learn which words in our vocabulary were the best indicators of relevance. We analysed the n -gram component of our compound feature vectors in order to calculate the *informativeness*, or *information gain* of each word unigram. The information gain of each feature is based on the prior probability of the feature pair occurring for each class label. A higher information gain, corresponding with a more informative feature, is obtained from a feature which occurs primarily in one class and not in the other. Similarly, less informative features occur evenly in both classes. The information gain idea is pivotal to the decision tree algorithm but generalises to others and was adapted in the NLTK package for use in a broader sense. In NLTK, informativeness of a word w was calculated as the highest value of $P(w = \text{feature_value} | \text{class})$ for any class, divided by the lowest value of $P(w = \text{feature_value} | \text{class})$ [95]. This informativeness I , is summarised below:

$$I = \frac{\forall c \in C : \max(P(\text{feature} = \text{feature_value} | c))}{\forall c \in C : \min(P(\text{feature} = \text{feature_value} | c))}$$

where C is the set of all classes and c is a possible class, and *feature_value* is a boolean indicating the presence or absence of that word.

Recall that to collect tweets, we make use of Twitter’s streaming API which allows us to specify keywords that restrict the data collection only to Tweets containing those specific terms. We try to measure the usefulness of the keywords we selected. To do this, we assess their information retrieval performance. Specifically, we used the precision-recall metric. In an information retrieval context, precision and recall are defined in terms of a set of retrieved documents and their relevance. We use our set of labelled tweets for this assessment. Here, the labelled tweets make up the set of retrieved documents and the tweets labelled as belonging to the “relevant” class make up the set of relevant documents. In this context, recall measures the fraction of relevant tweets that are successfully retrieved while precision

measures the fraction of retrieved tweets that are relevant to the query.

5.5.3 Generalization and Validation

We cannot use the same data that we use in building our models, to also test them. Instead, a separate test set is required for the purpose of evaluating the performance of the constructed model. The constructed model is then applied to each instance in the test set, and its performance evaluated by comparing its predictions to the actual known labels of the instances. In doing this, the separate test set acts as an approximation for new data, and allows us some insight on the generalization capabilities of our models.

We make use of a hold-out validation setup when building and evaluating our models. The labelled data set is randomly partitioned into training/validation/test splits with the ratios 65/5/30. The training set is used to build the models. The validation set is used to select hyperparameters for models where necessary. The test set is used for evaluating the models.

5.5.4 Correcting the Class Imbalance

27% of our labelled tweets were marked as relevant, while 73% was labelled as irrelevant. Imbalanced data causes well known problems to classification models [1]. We initially tried both oversampling and undersampling techniques to create a balanced training dataset, as well as, just using the unbalanced data. We found no major difference between the balancing approaches, but they gave some advantage over the unbalanced data, so we opted for over sampling. The class distribution over the balanced training set had 47% of tweets as relevant and 53% as irrelevant. The test set, however, was not balanced, and left as it was. This was done with the goal of our test set simulating the real application scenario as closely as possible.

5.5.5 Statistical Tests

It is sometimes useful to check whether the observed differences between the performances of two models is merely due to chance, or statistically significant. For this, we made use of the paired t-test.

Paired t-Test The paired t-test is used to determine whether an observed within-pair difference is larger than would be expected to have occurred by chance. The assumptions of the paired t-test are:

1. The data is continuous
2. The data follow a normal probability distribution.

3. The data is independent.

The paired t-test is calculated as:

$$t = \frac{\bar{d}}{\sqrt{\frac{s^2}{n}}}$$

where \bar{d} is the mean difference, s^2 is the sample variance and n is the sample size. The probability that an observed difference is a chance occurrence is given by the p-value obtained from the test.

Pearson Correlation Correlation analysis is typically used to test the strength and direction of the relationship between two variables. Pearson correlation, also known as *Pearson's r* or the *bivariate correlation*, is one such measure of the linear correlation between two variables. The Pearson correlation, ρ , is defined as:

$$\rho_{x,y} = \frac{E((x - \mu_x)(y - \mu_y))}{\sigma_x \sigma_y}$$

5.6 Summary

In building a syndromic surveillance system from Twitter data, we need to be able to extract a signal from the streamed Tweets. For this, we must process the collected data in order to extract the relevant content from the noisy Twitter data. In this chapter, we described the methodology involved in the relevance filtering used to de-noise the Twitter data and build a reliable signal. We proposed and describe a number of approaches to solving the relevance filtering problem. We looked towards techniques based on semi-supervised learning due to the fact that such techniques offer us the much appreciated benefit of getting the most out of our vast amounts of unlabelled data, without requiring any extensive labelling efforts. Additionally, as shown in chapter 3, we identified a gap in the literature with regards to semi-supervised approaches to syndromic surveillance despite its advantages and sought to explore this further. First, we investigated iterative labelling algorithms, making use of both self-training and co-training. We then looked towards techniques also based on deep learning. While doing this, we conceptualized and experimented with a novel deep learning algorithm which we termed the Generative Classification Network (or GCN for short). The GCN is a semi-supervised text classification algorithm that makes use of neural language models built on both labelled and unlabelled data to perform classifications. Next, we applied an attention-based bi-direction Recurrent Neural Network. We further distinguished between two variants of our proposed neural network architecture, based on the Long Short Term Memory and Gated Recurrent Unit architectures respectively,

termed the ABLSTM and ABGRU. In doing so, we combined the inherent pattern recognition capabilities of deep learning, with the identification and use of keywords that is typically employed by traditional classification methods. Finally, we also described the evaluation methodology used to analyse the performance of our proposed algorithms.

Chapter 6

Results of Relevance Filtering and Syndromic Surveillance

6.1 Introduction

Our proposed system is intended to collect and process Twitter data in real-time. However, we found that the collected data is very noisy. In order to obtain a reliable syndromic surveillance signal from the data, we attempted to denoise the data. We carried out this denoising by means of automatically filtering relevant and irrelevant Tweets. This relevance filtering was implemented through semi-supervised text classification. In this chapter, we present and discuss the results of our efforts to build a syndromic surveillance system from Twitter data. First, we present the results of our feature experimentation which was carried out to inform us on the best feature extraction routes and representations to apply to our data. Once this was determined, we implemented our proposed semi-supervised learning approaches to relevance filtering. We drew comparisons between the performances of our algorithms and other successful and popular approaches to text classification. For the sake of completeness, we also apply popular fully supervised techniques to the task of relevance filtering and compare our approaches to them.

6.2 Feature Analysis

We assessed the discriminative ability of each hand-crafted features in order to assess their utility, as well as ascertain if any of them hindered our per-

formance. This analysis was performed for both our hand-crafted features and our text embeddings, which are automatically learned from patterns in the data.

6.2.1 Hand-made Features

We evaluated the performance of these features using feature ablation experiments. We evaluated the performance of a given classifier (we chose the fully supervised Naive Bayes classifier for its simplicity) when using all our features, and then again after removing each one of these features. The difference in the performance is used as a measure of the importance of the feature. Our hand-crafted features are meant to be used on top of a traditional n -gram approach. Out of curiosity, we also performed the ablation experiment on the n -gram feature in order to understand its contribution. Table 6.1 shows the results of this experiment. As was expected, n -grams, are an effective and reliable feature, and serve as a good starting point to build additional features on top of. We found that our supporting features yield some additional improvements in performance on top of the n -gram features. For each of these supporting features, their omission results in a performance drop of around 0.1. Of our additional features, we found that *Negative Emojis/Emoticons* were the most discriminative, followed by the *Denotes Laughter* feature. Both of these features capture emojis in addition to colloquialisms. We also observed that *Contains Asthma-Verb Conjugate* and *Indicates Asthma Possession* underperformed compared to the other features. These features only contributed a margin of 0.02 and 0.01 respectively. Consequently, these features were discarded and not utilized in our systems.

Ablated Feature	F2 Score
<i>None</i>	0.714
<i>n-grams</i>	0.596
<i>Contains Asthma-Verb Conjugate</i>	0.690
<i>Indicates Asthma Possession</i>	0.693
<i>Denotes Laughter</i>	0.643
<i>Negative Emojis/Emoticons</i>	0.620
<i>Word Classes</i>	0.637
<i>Positive/Negative Word Count</i>	0.625

Table 6.1: F1 scores after feature ablation

Word	I (Relevant:Irrelevant)	Relevant Prior Probability	Irrelevant Prior Probability
chest	22/4	0.96	0.04
throat	17/1	0.95	0.05
wow	17/1	0.95	0.05
health	1/17	0.06	0.94
cold	16/1	0.94	0.06
moment	15/1	0.94	0.06
forecast	1/14	0.07	0.93
awake	13/1	0.93	0.07
awful	13/1	0.93	0.07
sick	13/1	0.93	0.07
cough	12/1	0.92	0.08
pollution	1/12	0.08	0.92
bed	11/1	0.91	0.09
hate	11/1	0.91	0.09
watch	10/1	0.91	0.09

Table 6.2: Most informative words measured by their *Informativeness* and their relevant:irrelevant prior probabilities

they contained said emoji. Overall, it can be seen that each of these emojis tends to lean heavily toward one class. This confirms that they can be quite discriminative and useful indicators of class membership and hence, helpful features.

6.2.2 Text Embeddings

Following our experiments around the hand-crafted features, We sought to determine which of our text embedding feature was best for our purposes. We tried not only to understand which word embedding algorithm performed best, also which method for building a representation for a sequence of words, was most appropriate for representing Tweets in our systems. To do this, we constructed Multilayer Perceptron (MLP) neural networks using





















Emoji	Occurrences for Relevant:Not- relevant Classes	Emoji	Occurrences for Relevant:Not- relevant Classes
	17:49		5:2
	31:9		6:1
	27:9		5:2
	21:12		3:2
	17:6		4:1
	11:6		3:1
	12:3		3:1
	10:3		3:0
	11:0		0:3
	8:2		2:1

Table 6.3: Most frequent emojis in labeled data and their distributions

Skipgram word vectors, CBOW word vectors, GloVe word vectors, PV-DM document vectors and PV-DBOW document vectors as feature representations of tweets. When using word vectors for feature representations of Tweets, we considered the feature vector of a Tweet to be the mean of the embeddings for the words in the tweet. Table 6.4 shows the results we observed. We found that taking the mean of the GloVe vectors of the words in a Tweet gave us the best performance. Because of this, we decided to use GloVe to represent words and Tweets in our experiments moving on.

6.3 Iterative Labelling Experimentation

6.3.1 Experiments and Results

We implemented both of our proposed semi-supervised iterative labelling algorithms - *self-training* and *co-training*, and applied them to our relevance filtering problem and dataset. We also applied a variety of popular and powerful supervised classification algorithms to the problem namely - Naive Bayes, Decision Trees, Logistic Regression, Support Vector Machines (SVMs) and Multilayer Perceptron (MLP) neural networks. We

Table 6.4: Classification performance of different Tweet feature representations obtained from deep embeddings

Tweet Embedding Algorithm	F-Measure	
Skipgram Mean	<i>Precision</i>	0.775
	<i>Recall</i>	0.720
	<i>F2</i>	0.732
CBOW Mean	<i>Precision</i>	0.675
	<i>Recall</i>	0.647
	<i>F2</i>	0.652
GloVe Mean	<i>Precision</i>	0.729
	<i>Recall</i>	0.765
	<i>F2</i>	0.757
PV-DM	<i>Precision</i>	0.588
	<i>Recall</i>	0.625
	<i>F2</i>	0.618
PV-CBOW	<i>Precision</i>	0.675
	<i>Recall</i>	0.718
	<i>F2</i>	0.708

used the Python implementations found in the Natural Language ToolKit (NLTK) and Sci-Kit Learn [96]. The results of our fully-supervised and semi-supervised classification are presented in table 6.5. Of the fully-supervised classifiers, Logistic Regression, SVM and MLP are very sensitive to hyper-parameters. The values for these hyper-parameters were found using grid-search with a hold-out validation setup. In the following evaluation, we use the discovered optimal hyper-parameters according to the grid-search. For Logistic regression, we used L2 regularisation with a regularisation strength C of 0.00001. We experimented with C within the range of $\{1e^{-5}, 10\}$ in steps of e^1 . For the SVM, we used a Radial Basis Function kernel and C of 0.01. We experimented with C within the range of $\{1e^{-5}, 1\}$ in steps of e^1 . For the MLP, we used 2 hidden layers, each with 128 neurons, a learning rate of 0.001, a regularisation α of 0.0001, a batch size of 200 and trained for 100 epochs. We experimented with learning rates within the ranges of $\{1e^{-5}, 1\}$ in steps of e^1 and α within the ranges of $\{1e^{-5}, 1\}$. The Adam optimiser [65] was used in minimising the loss function. For the iterative labelling experiments, we varied and tuned the confidence thresholds until

we found the best results and reported those. Below, we also discuss in more detail how the confidence threshold affected the iterative labelling performance as it is a key aspect of the algorithms. The best fully-supervised approach according to a combination of the F_1 and F_2 scores was the MLP, which achieved an F_2 score of **0.888** on the test data. This equated to an overall prediction accuracy of **95.5%**. The best semi-supervised approach, which was the co-training algorithm (using the best fully-supervised classifier - MLP as its base), achieved an F_2 score of **0.929** on the test data, also with a predictive accuracy of **95.5%**. Overall, our iterative labelling approach achieves higher F scores. To confirm what we concluded from the results, we applied a paired t -test to test the difference in F_2 scores between the fully-supervised MLP algorithm and the co-training algorithm. Before carrying out this test, we confirmed that the data satisfied the assumptions necessary for the paired t -test to be relevant - continuous, independent, normally distributed data without outliers. For the paired t -test, the train and test sets were concatenated to form the whole dataset and randomly split for each iteration of the test. This resulted in a t -statistic of 7.7 and a **p-value of 1.7×10^{-13}** which suggests that the difference between the F_2 scores of the two algorithms was not due to chance.

Supervised Algorithms	Precision	Recall	Accuracy	F_2 Score
NB	0.636	0.804	84.2%	0.764
DT	0.915	0.629	89.7%	0.671
RF	0.832	0.815	90.4%	0.818
LR	0.885	0.739	91.5%	0.764
SVM	0.864	0.722	90.6%	0.747
MLP	0.928	0.878	95.5%	0.888
Semi-Supervised Algorithms	Precision	Recall	Accuracy	F_2 Score
Self-training	0.897	0.924	95.6%	0.919
Co-training	0.881	0.942	95.5%	0.929

Table 6.5: Results of relevance classification on the test data. Naive Bayes (NB), Decision Tree (DT), Random Forest (RF) Logistic Regression (LR), Support Vector Machine (SVM) and Multilayer Perceptron (MLP) algorithms are reported together with the self-training and co-training iterative labelling algorithms.

To give a better understanding of how the different measures manage to balance the number of FP and FN, we also present the confusion matrices for both the best performing fully supervised and iterative labelling methods on the test data. These confusion matrices are shown in tables 6.6 and 6.7 respectively. From the confusion matrices, we see that the iterative labelling approach performs better for the purpose of syndromic surveillance as it yields only 17 false negatives even though it also yields 37 false positives. Considering that our aim is to develop a filtering system to identify the few relevant tweets in order to register a signal for syndromic surveillance, it is critical to have high recall, hopefully accompanied by high precision, and therefore high accuracy. The iterative labelling method is able to identify and retain relevant tweets more often, while also being able to identify irrelevant tweets to a reasonable degree. Hence, even with a shortage of labelled data, the iterative labelling algorithms can be used to filter and retain relevant tweets effectively.

		Actual Response		
		True	False	Total
Predicted Response	True	TP (256)	FP (20)	276
	False	FN (35)	TN (891)	926
Total		291	911	$N = 1202$

Table 6.6: Confusion matrix for MLP fully-supervised classification on the test data

		Actual Response		
		True	False	Total
Predicted Response	True	TP (274)	FP (37)	311
	False	FN (17)	TN (874)	891
Total		291	911	$N = 1202$

Table 6.7: Confusion matrix for Co-training iterative labelling algorithm on the test data

Figure 6.3.1 shows how the performances of our iterative labelling systems change as the confidence threshold changes. The confidence threshold controls how conservatively the iterative labelling system assimilates unlabelled

instances as it represents how confident the iterative labelling system needs to be in its classification before assimilating the instance to inform future decisions. We observed co-training with MLP to perform best. We also observed that for lower confidence thresholds between 0.1 and 0.5, self-training performance is usually lower and does not change much between thresholds. Co-training on the other hand, appears to be less sensitive to this parameter. Figure 6.3.1 also reiterates what we learned from table 6.5 that the MLP is our strongest fully-supervised model. In addition, while the logistic regression classifier does not perform as well as the MLP, it appears to be robust to different confidence thresholds when used in an iterative labelling context. We hypothesise that this advantage arises because the logistic regression classifier has considerably less hyper-parameters to optimise. This means that if a set of hyperparameters, which is impactful on performance, is not optimal for a certain threshold, such a set would be less of a hindrance to the logistic regression model.

The main issue with our iterative labelling approach is that, because the classifiers are not perfect and do not have 100% accuracy, we cannot be sure that the unlabelled instances that they label for assimilation are always correct. This means that their initial performance before any labelling iterations is vital. Consider a classifier, initially of poor performance (with an accuracy of 0.2 for example). When classifying unlabelled instance with which to train itself, 80% of its classifications will be wrong, so it will assimilate false hypotheses, which will in turn make its performance in the next iteration even worse and so on. Conversely, if the initial accuracy is high, it is more likely to correctly classify unlabelled instance and be less resistant to the drop in performance from assimilating false hypotheses. We conducted an experiment to measure the quality of the automatically labelled instances assimilated by our iterative labelling classifiers. For this exercise, we used the second set of labelled tweets from a different time period as the “unlabelled” set with to which the iterative labelling is applied to. The same training set as in our other experiments was used for the initial training stage. The self-training and co-training processes were initiated, applying these classifiers to the alternative set of labelled data (around 2000 instances) in steps of 200. Figure 6.3.2 shows a plot of the proportion of correctly classified instances that the iterative labelling process assimilated. The co-training approach had a higher rate of being correct when making new additions. This was in fact the aim of adopting co-training with its multiple different views of the same data. The proportion of correct assimilations of both the self-training and co-training methods rises as more data is assimilated, due to the fact that the systems are getting more intelligent. Although we could not test beyond 2000 instances (because of our limited labelled data), we believe that the proportion of correct assimilations will increase until a certain point, after which it will plateau. We expect this plateau due to the fact that at a certain point, the iterative learning classifiers will have nothing new to learn from new data after having been exposed

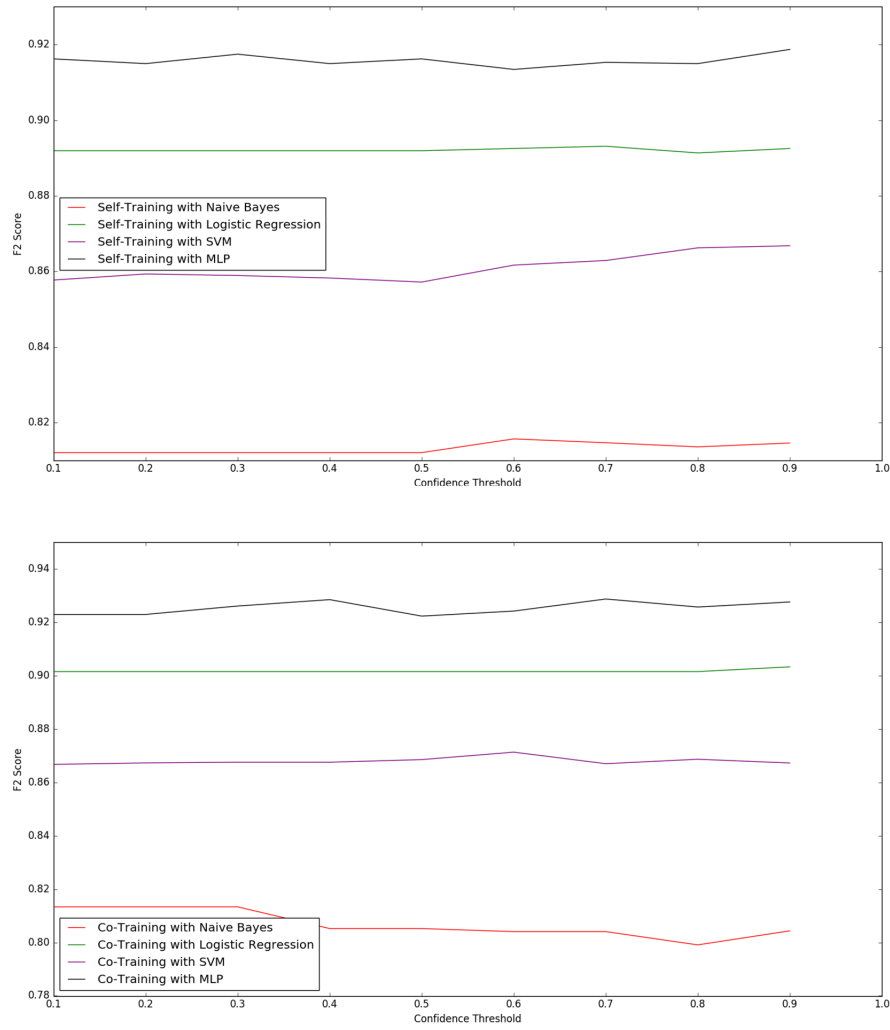


Figure 6.3.1: Graph of F2 performance of Iterative Labelling using different confidence thresholds

to so much.

6.3.2 Discussion

We made use of our initial labelled dataset from collection period 1 to assess the proficiency of our proposed semi-supervised iterative labelling algorithms. Using this algorithm for text classification in filtering Tweets, we achieved an accuracy of 95.5% and F_1 and F_2 scores of 0.910 and 0.929 respectively. We argue that recall is very important for us because we want to keep all the relevant Tweets so that we can have some signal, even if amplified by some misclassified irrelevant Tweets. The best recall, obtained by the co-training algorithm, equated to retaining over 90% of the relevant

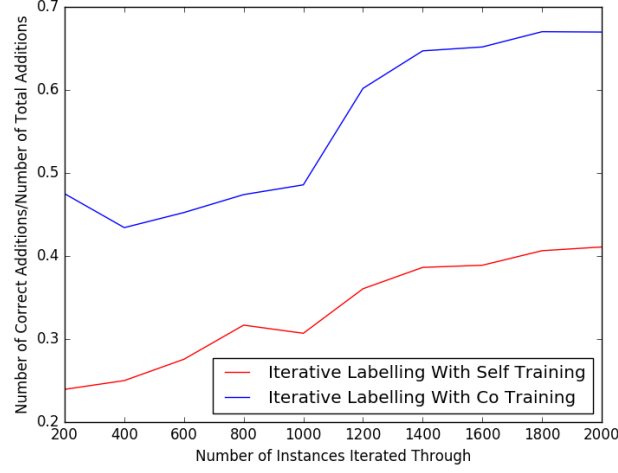


Figure 6.3.2: Graph showing how many correct assimilations the iterative labeling algorithms make per iteration using labelled data from a different time period

tweets after classification. Also, due to the semi-supervised nature of the proposed algorithm, we were able to use 8000 previously unlabelled Tweets before it started to see a deterioration in performance. This allowed us to make more use of the data collected.

In terms of training and inference speeds, the iterative labelling algorithms are somewhat interesting, with them being meta-algorithms which wrap around other algorithms. As such, their time complexity is closely dependent on that of their seed algorithms. Self-training and co-training are relatively slow to train as their training time complexity are dependent on not only the training time complexity of their seed algorithms, but also on the inference time complexities. The best and worst case training time complexities of an iterative labelling algorithm f , with an unlabelled dataset U , a labelled dataset L , a seed algorithm A , and a *CHOOSE-LABEL-SET* sample size, k , are shown in equations 6.3.1 and 6.3.2 respectively.

$$f(U, L, A) = \Theta((\Theta(\text{TRAIN}_A(U)) + \Theta(\text{INFERENCE}_A(k)))) \quad (6.3.1)$$

$$f(U, L, A) = \Theta((\Theta(\text{TRAIN}_A(U)) + \Theta(\text{INFERENCE}_A(k))) \times \frac{U}{k}) \quad (6.3.2)$$

In the best case, there is only one training iteration, and the seed algorithm A is trained once, used to relabel and the stopping condition is met. In the worst case, the stopping condition is not met until all of the unlabelled data is iterated through. The choice of seed algorithm is a very important consideration when using iterative labelling algorithms as the seed algorithm plays a big part in the overall time complexity. The inference time complexity of an iterative labelling algorithm is exactly equal to the inference

time complexity of its seed algorithm.

6.4 Generative Classification Network Experimentation

6.4.1 Experiments and Results

We now look to deep learning approaches to solving the relevance filtering problem for syndromic surveillance. We implemented our experimental Generative Classification Network (GCN) model, which is a model based on deep generative neural language models. As described in chapter 4, we underwent a number of labelling efforts over the course of the project. Our GCN was implemented and evaluated using the expanded labelled set of size 5000. For the sake of comparison, we also experimented with popular deep learning algorithms which have seen wide success in text classification tasks. We made use of the Multilayer Perceptron (MLP) model, Convolutional Neural Network (CNN) model and Long Short Term Memory Recurrent Neural Network (LSTM RNN) models. We made use of the text classification CNN introduced by Kim [135] and the short-text classification RNN by Nowak et al. [193]. We compared the results obtained using these models to those achieved by our proposed GCN model. We present the results of this experiment in table 6.8. Note that all our results were computed from the test partition. The experimental GCN outperformed the CNN at the task of relevance filtering. However, we found that the LSTM classifier performed best, outperforming both the GCN and CNN and yielding the highest F_2 score, our preferred measure.

When evaluating our deep learning approaches, we also considered the time taken to perform the relevance filtering. We measured and plotted the times taken for the GCN, MLP, LSTM and CNN to perform the relevance filtering, varying the number of Tweets they were fed up to 10,000 Tweets. For this experiment, we used **unlabelled** Tweets from the second collection period June 21, 2016 - August 30, 2016. This plot is shown in figure 6.4.1. From the plot, we can see that the LSTM RNN takes the most time while the MLP takes the least time. The GCN takes considerably less time than the LSTM RNN and CNN. This is likely due to the fact that the GCN doesn't use the neural network for the actual act of classification. Recall that instead, it generates what it thinks is a relevant and irrelevant Tweet using the first k words in a query Tweet and uses a distance measure to make its classification. With this in mind, the GCN understandably takes up less time in its operation than the other complex deep classification models. The MLP model is also a very simple model which does not take a lot of time to classify Tweets. While the MLP beats the GCN in time, the

Deep Classifier	F-Measure	
Multilayer Perceptron	<i>Precision</i>	0.729
	<i>Recall</i>	0.765
	<i>F2</i>	0.757
Convolutional Neural Network	<i>Precision</i>	0.521
	<i>Recall</i>	0.779
	<i>F2</i>	0.709
Recurrent Neural Network (LSTM)	<i>Precision</i>	0.638
	<i>Recall</i>	0.841
	<i>F2</i>	0.791
Generative Classification Network	<i>Precision</i>	0.500
	<i>Recall</i>	0.800
	<i>F2</i>	0.714

Table 6.8: Performance of Generative Classification Network with baselines on relevance filtering task.

GCN yields better results than the MLP. We also observed that the time

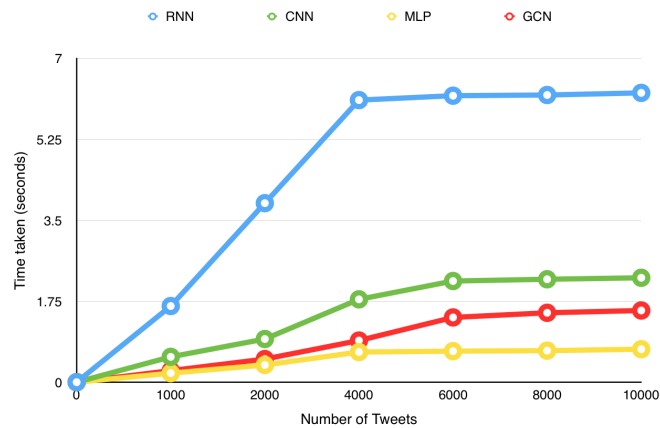


Figure 6.4.1: Time taken to perform relevance classification on a collection of Tweets.

taken for relevance filtering rises steadily with the number of Tweets up until about 4000 Tweets. After this, the time taken changes very little as the number of Tweets rises. This is due to the fact that all Tweets get classified at once (at the cost of increased memory usage) by making use of the batch processing of TensorFlow. In the cases with 4000 Tweets and above, it would appear that the computer could not manipulate all of the

data together at once with its available RAM, so larger ROM or swap space is used which eliminates the need for incremental processing (as more space is available in that scenario). Nonetheless, this does not change the fact that the different neural networks spend different amounts of time on the relevance classification, despite the RAM or ROM memory conditions. The bulk of the difference in time spent on classification is down to the architecture of the network and the amount of setup required. From figure 6.4.1, we find that the relatively simple architecture of the MLP performs much quicker than that of the RNN, CNN and GCN. Furthermore, the RNN sees more drastic jumps in time taken for relevance filtering as the number of Tweets increase than the GCN or MLP.

6.4.2 Discussion

We found that the LSTM performed best, yielding the highest F_2 score, our preferred measure. RNNs take advantage of the sequential nature of text which is also exhibited by Tweets (which are short-texts). CNNs on the other hand are good at extracting position-invariant features in space. Because of the short nature of Tweets, even when they are represented in 2D space, CNNs do not have a lot of salient spatial information to work with and are outperformed by even the MLP as well. While we found our proposed GCN was outperformed by the LSTM, it still achieved better results than the CNN at the short-text classification task of relevance filtering.

It is also worth noting that language models are usually trained on large amounts of data while those used in our GCN were built using a comparatively small dataset. For reference, the models used in the Stanford GloVe experiments were built on crawls of Wikipedia yielding billions of tokens and a vocabulary consisting of millions of unique words [207]. In comparison, the language models in our GCN were built using a collection of 5,000 tweets each with a maximum length of 140 characters. We are currently limited in this way because the GCN requires labels along with the data it is built on. Considering this, we believe that the GCN might show some promise in a setting with lots more data. Having only introduced the GCN, we intend to perform further investigations with the GCN using standard data sets for text classification.

6.5 Attentive Bi-directional Recurrent Neural Network Experimentation

6.5.1 Experiments and Results

We evaluate the performance of our attention-based Bi-RNN for relevance filtering. We assess our proposed approach’s ability to automatically classify Tweets as “relevant” or “irrelevant” based on whether they associate with an individual expressing concern or discomfort over asthma/difficulty breathing or its symptoms. In these experiments, we compare the classification ability of our proposed approach to that of existing successful and popular approaches. Again, we make use of the text classification CNN introduced by Kim [135] and the short-text classification RNN by Nowak et al. [193] as baselines for our comparisons. As described in chapter 4, we underwent a number of labelling efforts over the course of the project. Our experiments were carried out using the expanded labelled set of size 8000. We implemented and applied both the *ABLSTM* and *ABGRU* flavours of our proposed algorithm to the Tweet relevance classification task. After applying our proposed algorithms and the baseline algorithms to the relevance filtering task and dataset, we observed the results presented in table 6.9. Note that all our results were computed from the test partition.

We found that the attentive RNNs outperformed the other architectures, with the ABLSTM being the stronger attentive RNN. As shown in chapter 5, the gating mechanism used by the GRU is smaller and less complex than that of the LSTM. This means that ABGRU is faster but not quite as accurate as the ABLSTM. The LSTM RNN was seen to achieve a higher precision than the ABLSTM and ABGRU but it fell behind in terms of recall. Its recall was quite low and negatively impacted its overall performance. In effect, this translates to it being more likely to find negative class examples, which were the majority class in the dataset. This suggests that it may be more suited to balanced datasets. However, our task of syndromic surveillance using Twitter deals with highly unbalanced data as most Tweets are not about health reporting. We also observed that the text CNN scored the worst in every metric, and as such, performed quite badly at the Tweet relevance classification, even though it had performed well at other text classification tasks [135].

When we described our attentive bi-directional Recurrent Neural Network In chapter 5, we communicated that the output of the attention layer is a Tweet attention vector, t . This vector summarizes the input word vectors while putting emphasis on important words. t is subsequently used as a vector representation for the Tweet in the classification part of the model. As such, the described model could also be applied to documents in other

Classifier	Metric	
<i>ABGRU</i>	<i>Accuracy</i>	0.900
	<i>Precision</i>	0.734
	<i>Recall</i>	0.656
	<i>F2</i>	0.666
<i>ABLSTM</i>	<i>Accuracy</i>	0.906
	<i>Precision</i>	0.752
	<i>Recall</i>	0.672
	<i>F2</i>	0.687
Convolutional Neural Network (CNN)	<i>Accuracy</i>	0.850
	<i>Precision</i>	0.507
	<i>Recall</i>	0.562
	<i>F2</i>	0.550
Recurrent Neural Network (LSTM)	<i>Accuracy</i>	0.889
	<i>Precision</i>	0.762
	<i>Recall</i>	0.557
	<i>F2</i>	0.589

Table 6.9: Performance of Attentive Bi-directional Recurrent Neural Network and baselines on Tweet relevance classification task.

problems to create meaningful embeddings for them.

To test this, we collected a random sample of Tweets, computed their attention vectors and performed t-distributed stochastic neighbour embedding (t-SNE) [165] dimensionality reduction to reduce their dimensions to 2. We then plotted these 2D attention vectors, shown in figure 6.5.1, in order to spatially visualize them. We found that Tweets with similar meanings and words appeared to be clustered together. In fact, in figure 6.5.1, it is possible to draw a decision boundary line that roughly separates both classes. This line is shown in red. Below the red line, we see Tweets which are symptomatic of the asthma/difficulty breathing syndrome. Above the line, we see Tweets which may contain keywords related to asthma/difficulty breathing but are not expressing concern or suffering. It is also worth noting that “*wheezing*” is often used as slang to exaggerate laughter. Twitter contains a lot of slang. The Tweet attention vectors capture the semantics of the different contexts of slang words, such as “*wheezing*”, and this boosts its

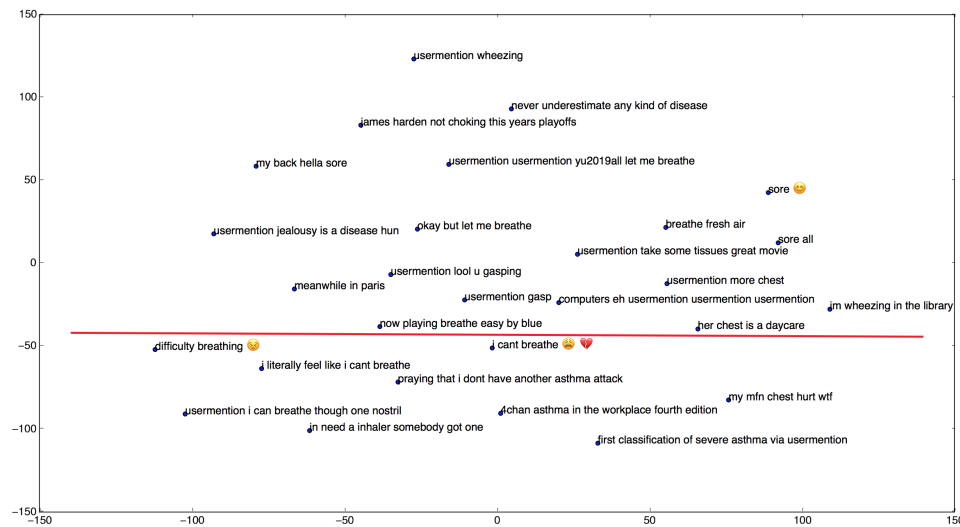


Figure 6.5.1: Plot of TWEETs representative of distances in attention embedding space. The axes represent t-SNE dimensional values.

discriminatory ability. The attention vectors give us a semantic and discriminatory vector representation for our TWEETs. As such, in addition to its utility for short text classification, our attentive model has the added ability to create useful document embeddings.

6.5.2 Discussion

We find from the literature that most Neural Network models used to classify TWEETs treat all words as equal while focusing on making use of semantic relationships between words to get the overall meaning. Our proposed attentive bi-directional RNN approach takes this a step further by not only trying to employ these semantic relationships, but also acknowledging the presence of key words and capitalizing on them. We experimented with LSTM and GRU units for the cells in our attentive bi-directional RNN. The attentive bi-directional LSTM (ABLSTM) approach was found to outperform the popular text-CNN and LSTM at the task of TWEET relevance classification.

As a demonstration, figure 6.5.2 shows a sample TWEET with our attentive

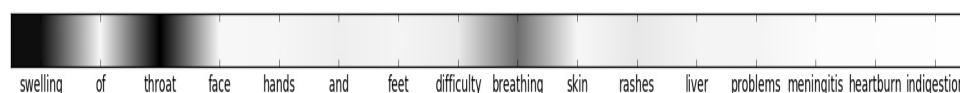


Figure 6.5.2: Heatmap showing weights placed on words in a TWEET by our attentive bi-RNN model

bi-RNN model applied to it. It shows the results of the attention layer. The

darker areas/words represent words which the model deems key to the message of the Tweet. We can see that the model does a good job of recognizing that *swelling*, *throat* and *difficulty*, *breathing* are important for determining whether the Tweet is relevant to our health context. We learned that the ABLTSTM model has strong understanding capabilities that can not only be used for accurate relevance filtering, but could also be taken advantage of for building informative document embeddings.

6.6 Syndromic Surveillance

While we have proposed and experimented with various approaches to text classification for relevance filtering, we would like to understand their potential utility for the generation of signals for syndromic surveillance. After constructing and evaluating our semi-supervised filtering systems, we assessed their utility for syndromic surveillance purposes by retrospectively applying them to Twitter data, and comparing the results against data for existing syndromic indicators from the Public Health England (PHE) Real-time Syndromic Surveillance Service. For this experiment, we made use of unlabelled tweets from our second collection period, June to August 2016. We performed comparisons with identified relevant anonymised data from PHE’s syndromic surveillance systems for this time period. PHE syndromic surveillance systems use primary care (general practitioner in hours and out of hours) consultations, emergency department (ED) attendances and tele-health (NHS 111) calls. For this analysis, a number of ‘syndromic indicators’ monitored by PHE’s syndromic surveillance systems were selected based upon their availability, quality and potential association to asthma/difficulty breathing. These indicators were “*difficulty breathing*” and “*asthma/whoeeze/difficulty breathing*”. As a control and sense check, we also compared our detected Twitter signal time series against non-respiratory syndrome data in the form of “*diarrhoea*” data.

6.6.1 Difficulty Breathing

The *Difficulty breathing* syndrome was generated from NHS 111 calls where callers made complaints about difficult or laboured breathing specifically. Similarly, the control *diarrhoea* syndrome data was also generated from NHS 111 calls. Daily counts of NHS 111 calls for *difficulty breathing*, together with daily counts of overall consultations were used to compute daily proportions of syndrome prominence. Similarly, for our Twitter systems, we employed our relevance filtering models to compute daily proportions of Tweets that were relevant to the *difficulty breathing* syndrome, relative to the number of Tweets collected each day. The resulting time series are shown in figures 6.6.1 - 6.6.5.

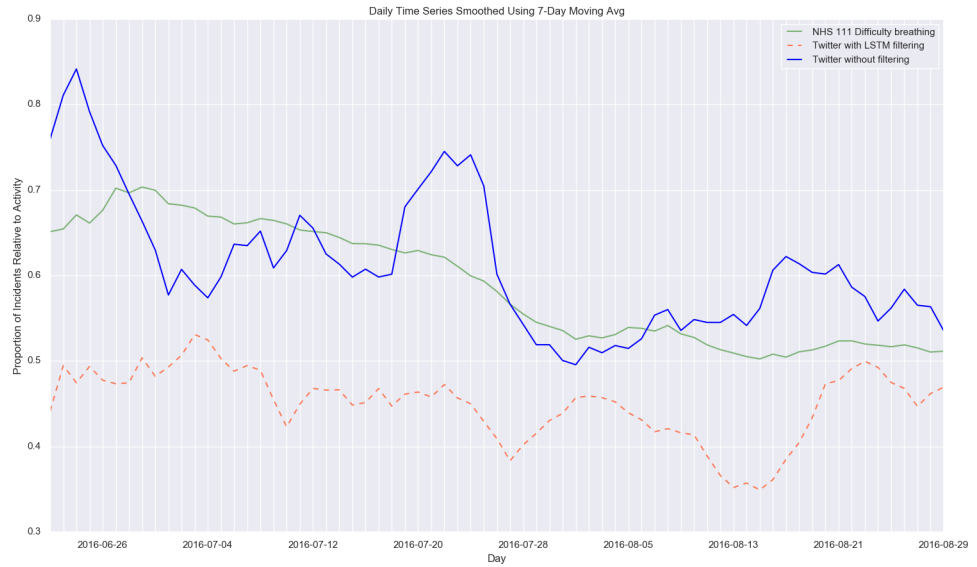


Figure 6.6.1: Comparison for Twitter signal extraction using LSTM relevance filtering

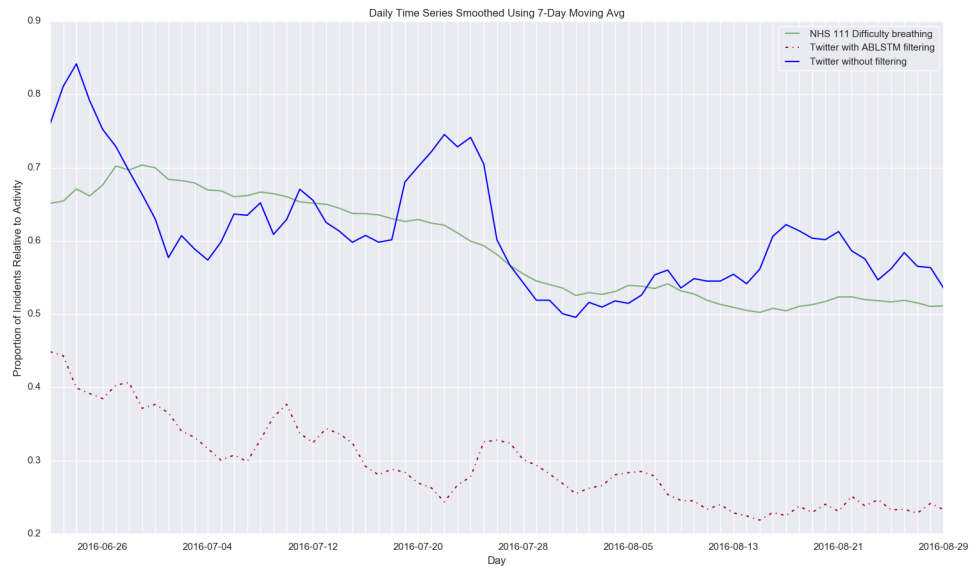


Figure 6.6.2: Comparison for Twitter signal extraction using ABLSTM relevance filtering

6.6.2 Asthma/Difficulty Breathing/Wheezing

The *Asthma/Difficulty Breathing/Wheezing* syndrome was generated from NHS GP Out-of-hours (GPOOH) consultations. This syndrome is a mix of different symptoms and point to the general affliction of respiratory disease. Again, the control *diarrhoea* syndrome data was also generated from NHS

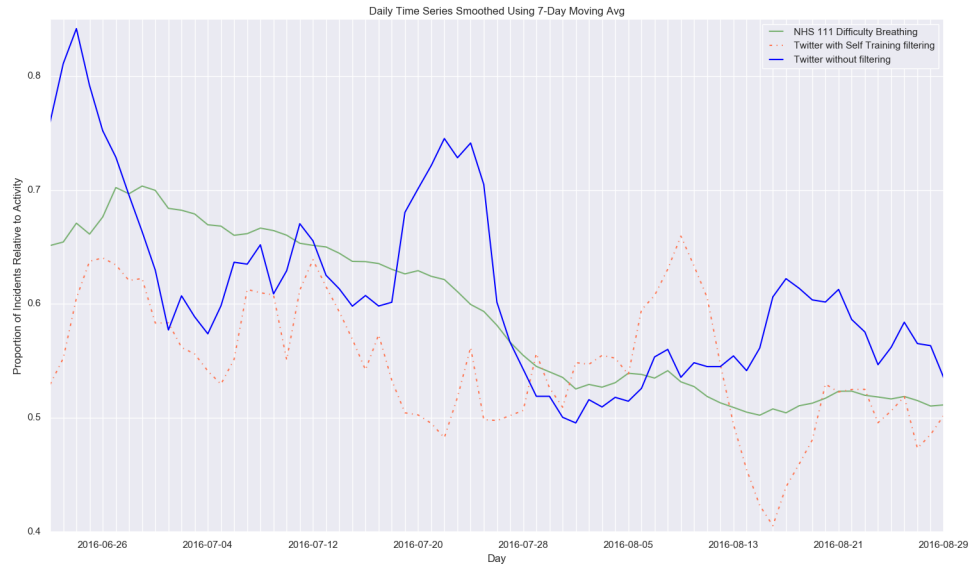


Figure 6.6.3: Comparison for Twitter signal extraction using Self Training relevance filtering

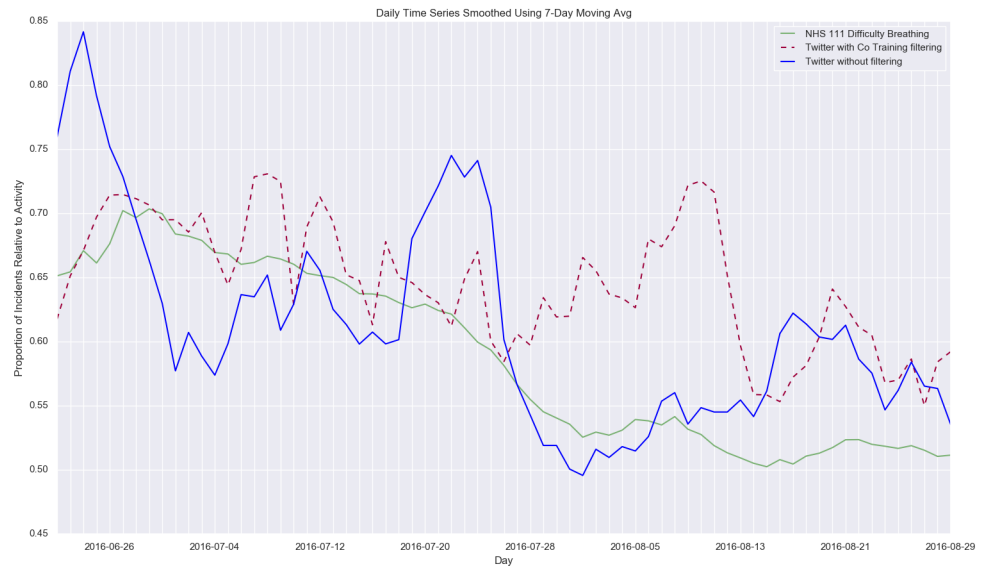


Figure 6.6.4: Comparison for Twitter signal extraction using Co-Training relevance filtering

111 calls. Daily counts of GPOOH consultations for *asthma/wheeze/difficulty breathing*, together with daily counts of overall consultations were used to compute daily proportions of syndrome prominence. Similarly, for our Twitter systems, we employed our relevance filtering models to compute daily proportions of Tweets that were relevant to the *asthma/wheeze/difficulty*

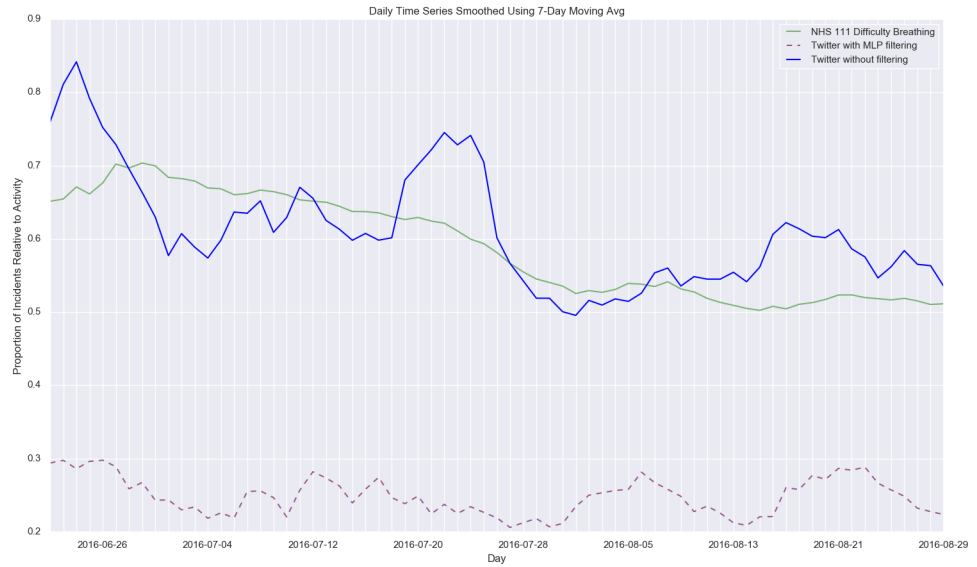


Figure 6.6.5: Comparison for Twitter signal extraction using MLP relevance filtering

breathing syndrome, relative to the number of Tweets collected each day. The resulting time series are shown in figures 6.6.6 - 6.6.12.

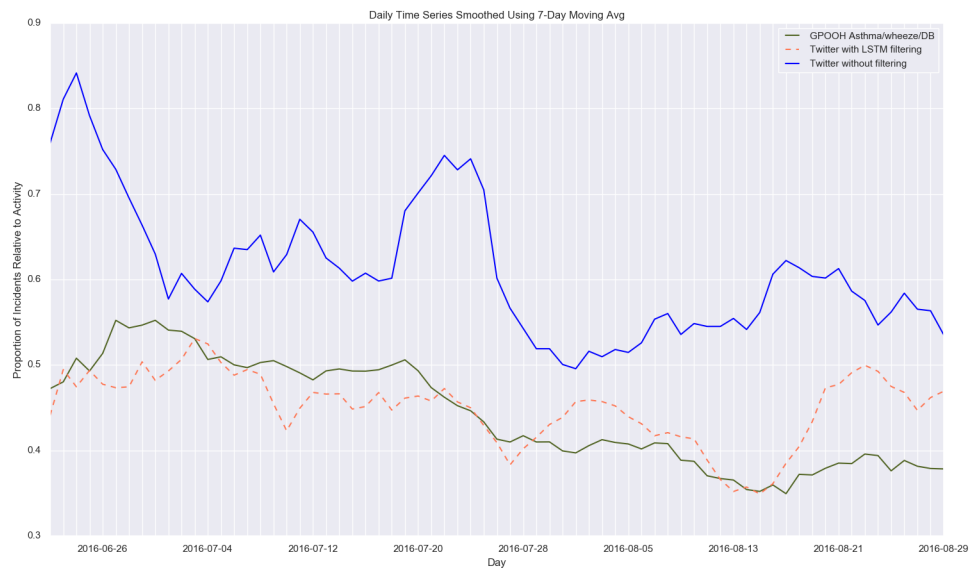


Figure 6.6.6: Comparison for Twitter signal extraction using LSTM relevance filtering

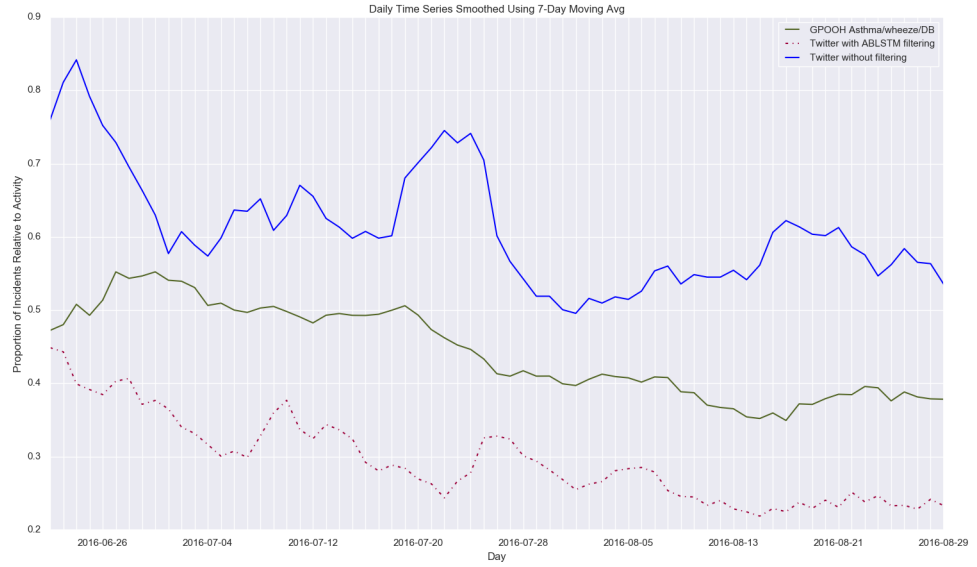


Figure 6.6.7: Comparison for Twitter signal extraction using ABLSTM relevance filtering

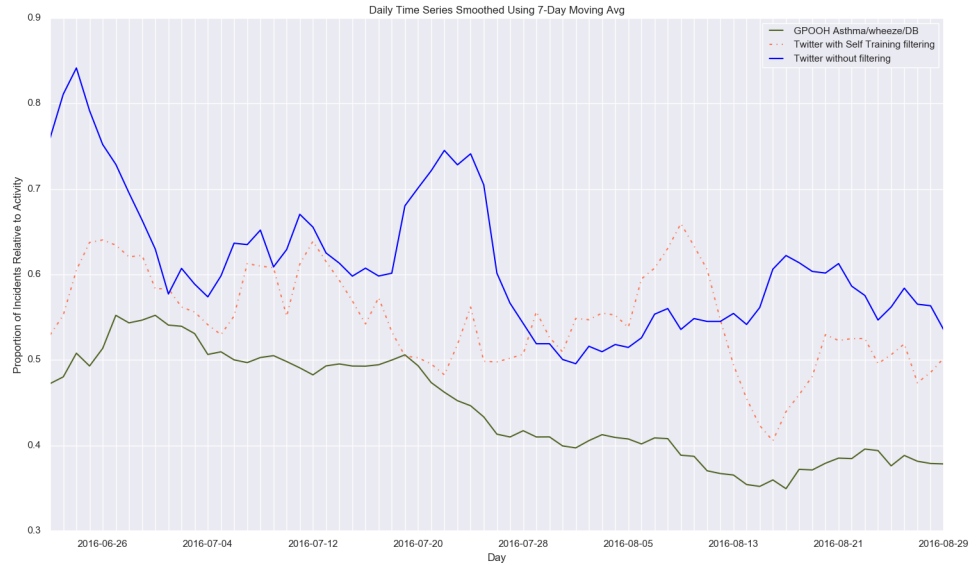


Figure 6.6.8: Comparison for Twitter signal extraction using Self Training relevance filtering

6.6.3 Control Syndrome: Diarrhoea

In this section, we compare the outputs of Twitter syndromic surveillance using our relevance filtering algorithms for asthma and difficulty breathing related Tweets, to recorded public health diarrhoea signals. These comparisons are shown in figures 6.6.11 and 6.6.11. Like with our research

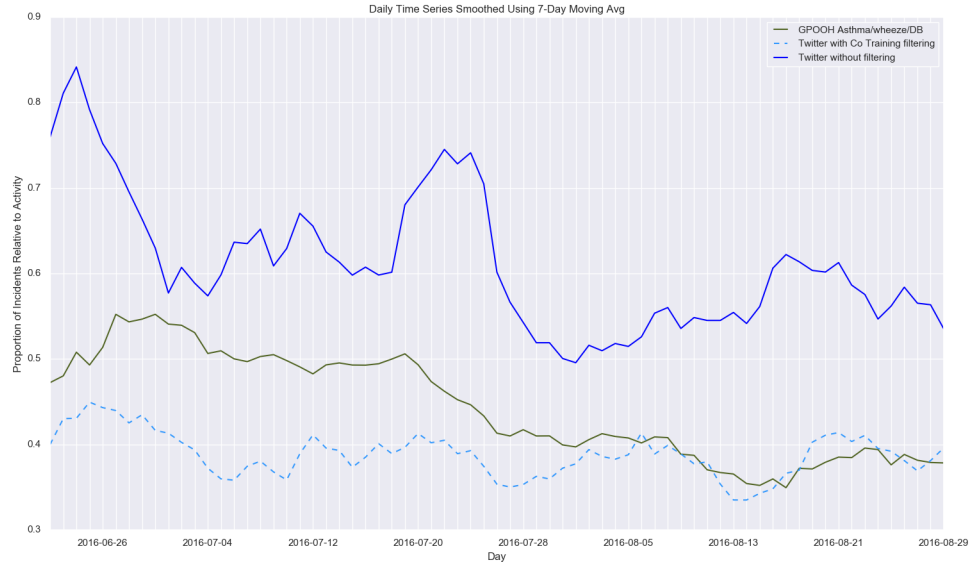


Figure 6.6.9: Comparison for Twitter signal extraction using Co-Training relevance filtering

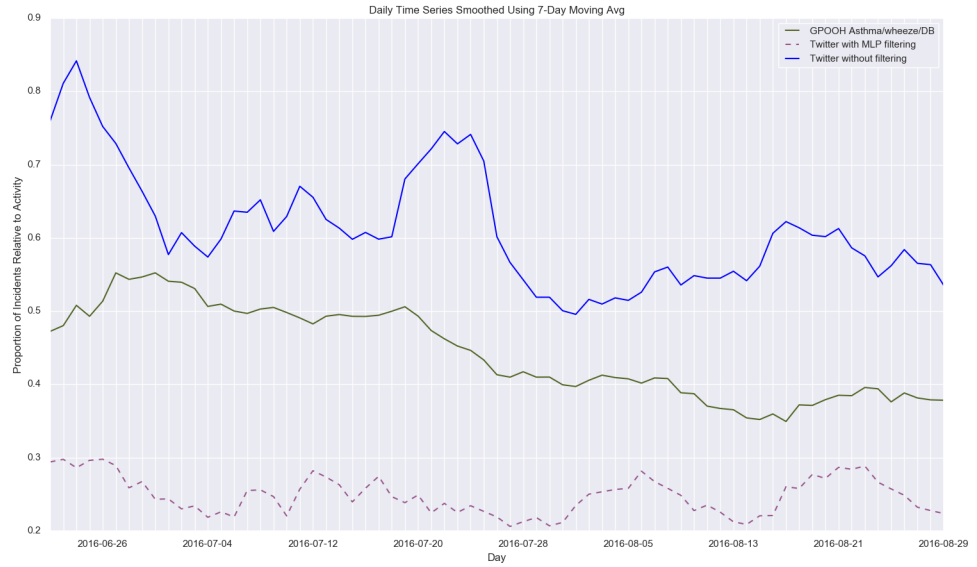


Figure 6.6.10: Comparison for Twitter signal extraction using MLP relevance filtering

signals, we used daily counts of NHS 111 calls for *diarrhoea*, together with daily counts of overall consultations to compute daily proportions of the diarrhoea syndrome prominence.

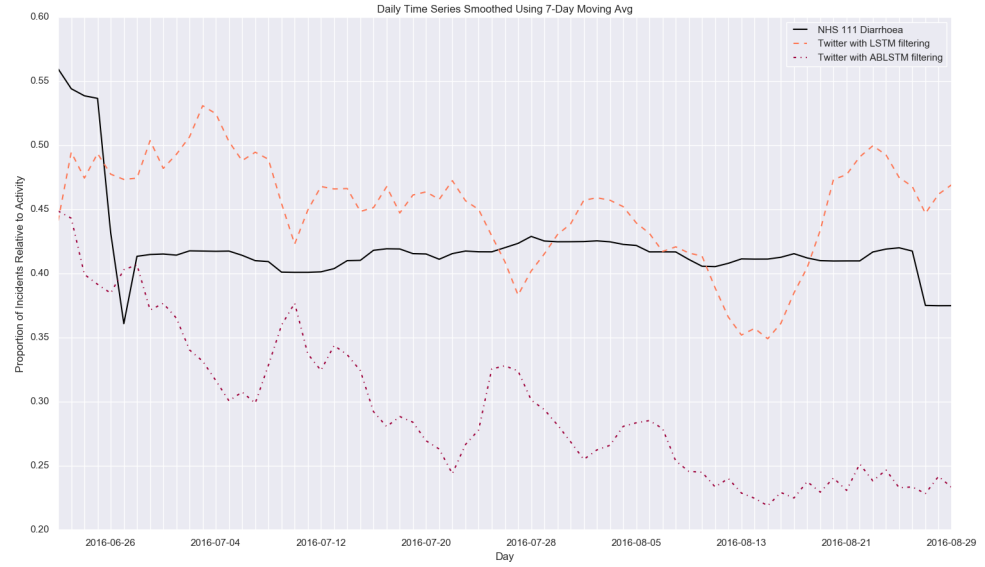


Figure 6.6.11: Comparison of Deep Learning Twitter signal extractions with control signal

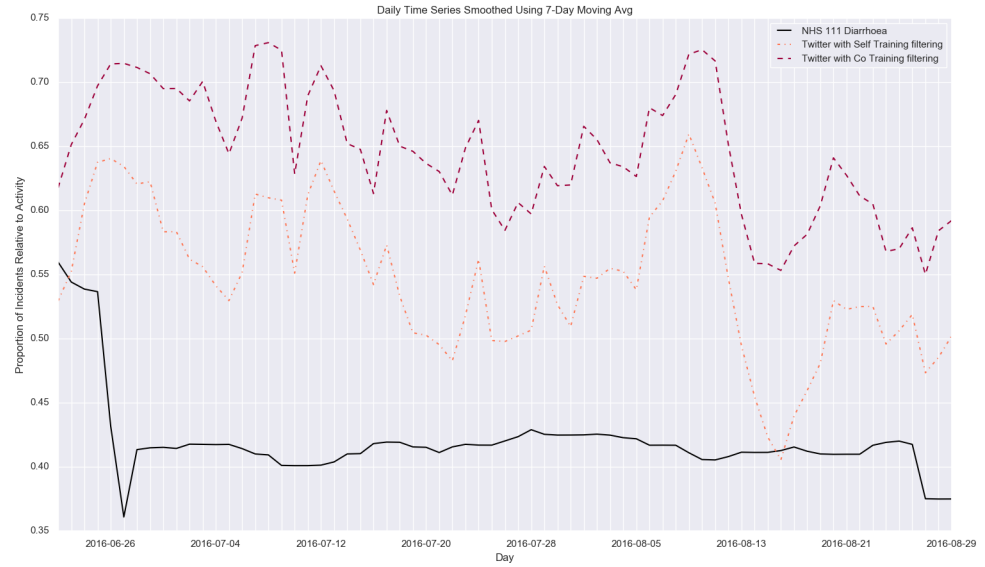


Figure 6.6.12: Comparison of Iterative Labelling Twitter signal extractions with control signal

6.6.4 Discussion

Figures 6.6.4 and 6.6.9 show the performance of our best iterative labelling algorithm, the co-training classifier. Figures 6.6.1 and 6.6.6 show the performance of the LSTM RNN found to outperform our Generative Classification Network (GCN). Figures 6.6.2 and 6.6.7 show the performance of

our ABLSTM. We also show a time series for the Twitter system without any relevance filtering. To do this, we took the daily counts of collected Tweets and normalised each day's count by the average Tweet count for that week. We smoothed the time series signals using a 7-day average to minimise the irregularities caused by the differences between weekend and weekday activities for GP out-of-hours services. In every figure, we see that the unfiltered Twitter signal is noisy and does not fit well with the signals for *asthma/wheeze/difficulty breathing* and *difficulty breathing*. Conversely, once our relevance filtering models have been applied, the Twitter signals follow a more similar shape and trend to the *asthma/wheeze/difficulty breathing* and *difficulty breathing*. The signal for *diarrhoea*, shown in figures on the other hand, does not appear to be related to any others as we may expect.

To gain a clearer picture of how well the signals matched, we performed some correlation analysis on them. We calculated the Pearson correlation coefficient to determine the strength and direction of any monotonic relationship between the indicators and our signals extracted from Twitter (table 6.10). Between the signal obtained by means of co-training and the

Relevance Filtering Algorithm	Syndrome		
	Asthma/Wheezing/DB	Difficulty Breathing	Diarrhoea
Co-Training	0.255($p = 0.03$)	0.214($p = 0.07$)	0.05($p = 0.7$)
MLP Neural Network	0.414($p = 0.0004$)	0.424($p = 0.0002$)	0.04($p = 0.7$)
LSTM RNN	0.637($p < 0.001$)	0.586($p < 0.001$)	0.125($p = 0.3$)
ABLSTM	0.792($p < 0.001$)	0.830($p < 0.001$)	0.207($p = 0.09$)

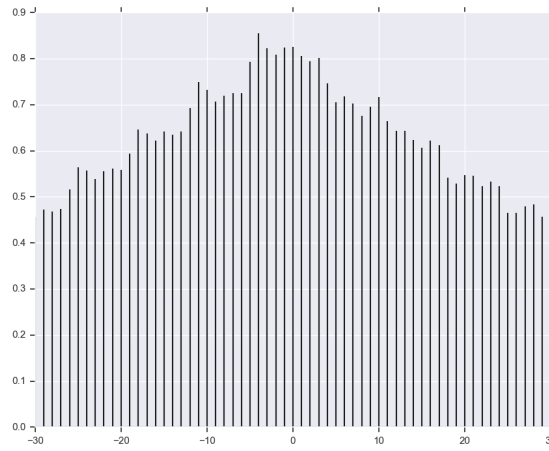
Table 6.10: Pearson correlations and P-Values for detected signals with syndromic surveillance signals

public health data, we observed a weak but statistically significant signal ($r = 0.424$). That being said, the signal produced by the co-training algorithm possessed a stronger correlation than that the signal produced by the best fully-supervised method examined in the iterative labelling experiments - the MLP classifier. The signal produced by the LSTM RNN, which beat our GCN model yielded a moderate statistically correlation with the public health data ($r = 0.586$). Finally, the signal obtained using the ABLSTM model has a strong and statistically significant correlation with the public health data ($r = 0.830$).

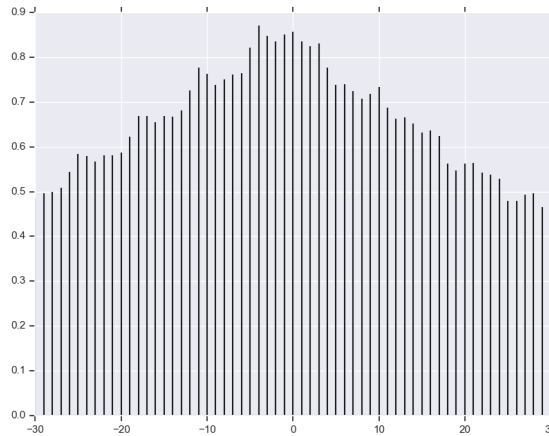
It is worth thinking about some potential causes of deviation of the Twitter signal from the ground truth. We identified two possible reasons for this:

- The Twitter signal could either be behind or ahead of the ground truth

signal. To investigate the first point, we performed some lagged correlation on the best Twitter signal, obtained using the ABLSTM and the ground truth *asthma* and *asthma/difficulty breathing/wheezing* signals. We computed the cross correlation between the raw counts of daily incidents identified on Twitter and the counts of reports from the public health data, varying the lag to produce the correlograms shown in figures 6.6.13a and 6.6.13b. These figures show that the highest cross correlation is observed at a lag of -4. This suggests that the Twitter signals are 4 days ahead of the ground truth data and might be a predictor of the ground truth.



(a) Cross correlation of ABLSTM Twitter signal with *asthma/difficulty breathing/wheezing* signal



(b) Cross correlation of ABLSTM Twitter signal with *difficulty breathing* signal

- The Twitter signal could be inadvertently picking up additional unforeseen events taking place at the time. For example, we noticed that in the time series figures, the raw Twitter signal peaked around the

date 20/06/2018. In order to better understand the reason for this spike, we examined the Tweets collected by our Twitter syndromic surveillance system around this period and found that a large proportion of them were Tweets like “Its too hot i cant breathe”. We then looked towards the historical meteorological data for that period, obtaining publicly available data from the London Met Office. This chart can be seen in figure 6.6.14. According to the London Met Office data, the hottest day in that period was around the date 20/06/2018, the same day we observed the spike in our Twitter signal. From this we learn that syndromes may not be clearly and

UK statistical summary

Mean Temperature

The mean value is 14.9 °C, which is 0.6 °C above the 1981-2010 average.

Rainfall

The total is 260 mm, which is 108% of the 1981-2010 average.

Sunshine

The total is 475 hours, which is 94% of the 1981-2010 average.

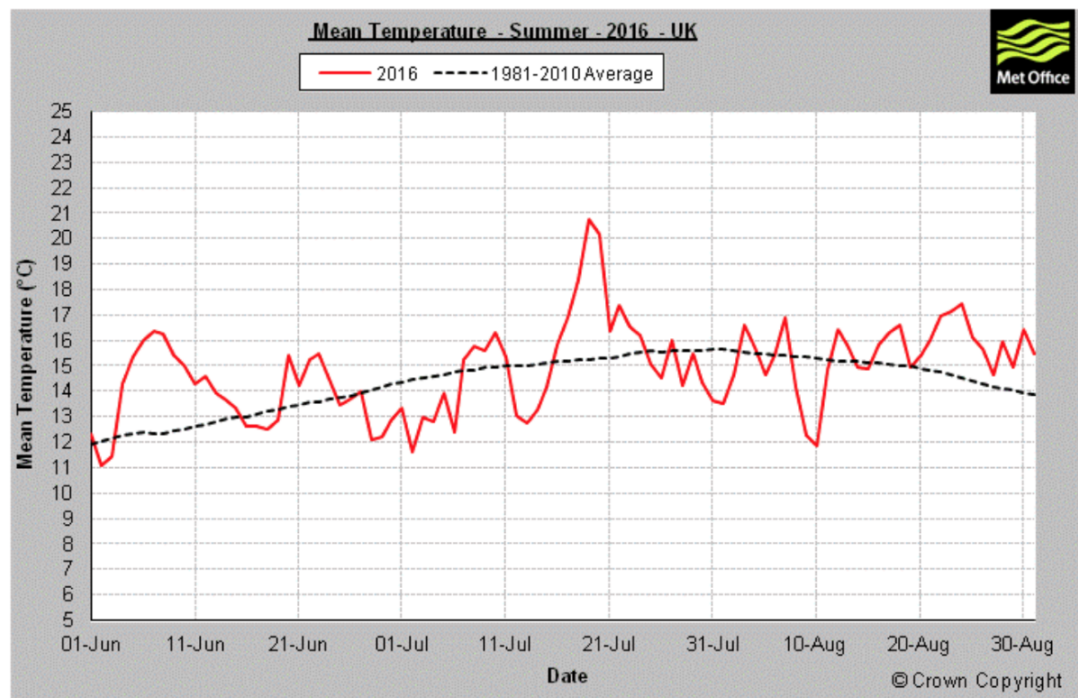


Figure 6.6.14: Chart of mean UK temperature for the time period of summer 2016 ¹

systematically expressed by reporters on Twitter, so our signals may be slightly perturbed by related events, such as meteorological events, causing difficulty breathing.

¹<https://www.metoffice.gov.uk/research/climate/maps-and-data/summaries/index>

6.7 Summary

In previous chapters, we introduced and proposed our ideas for effectively and efficiently obtaining a reliable syndromic surveillance signal from Twitter. In this chapter, we evaluated and assessed each of our proposed ideas. We introduced some novel features extraction techniques for Twitter data based around emojis and found them to work well. Upon evaluation, we found that our proposed Generative Classification Network (GCN) network did not perform as well as existing, more popular techniques. We argue that this is due to the fact that the GCN model is a data-hungry technique and insufficient data was used. Further experimentation on a standard text classification dataset is required to verify its general profitability. However, it was found inadequate for the task of relevance filtering. Our proposed attentive bi-directional RNN proved effective at the task of relevance filtering, beating out popular, widely successful deep learning techniques. Finally, we sought to evaluate the signals generated by each of our proposed relevance filtering algorithms. We found a strong statistically significant correlation between the signal generated by our attentive bi-directional RNN.

Chapter 7

Optimizing the Twitter Syndromic Surveillance Stream: Intelligent and Automatic Keyword Selection for Twitter Streaming

7.1 Introduction

In investigating the use of Twitter data for syndromic surveillance purposes, we have discussed the collection and effective filtering of this data in order to obtain a strong signal from it. We have studied in detail, the use of various statistical and machine learning algorithms to understand and filter Tweets relevant to a syndrome of interest. This objective is made doubly necessary when Tweets obtained from the Twitter stream through the official API are loosely filtered.

Recall from chapter 4, that we collect Tweets using the API in a real-time stream which can be filtered by location and content. The API allows provides the ability to filter Tweets based on content by specifying a set of keywords. Only Tweets matching the keywords are passed from the stream by the API. Choosing the right set of keywords can have a big impact on the syndromic surveillance system as it controls which Tweets our system collects. Choosing keywords that are too precise and strict will result in our system collecting mostly relevant Tweets, but also simultaneously only observing few Tweets, which will most likely only be a small sample of the relevant Tweets available. Conversely, choosing keywords that are too broad will result in our system observing a great deal of Tweets, most of which will not be relevant. Also, if the right keywords are used to initially stream the Tweets, there is less burden on the classification system used for

subsequent filtering. Therefore, selecting the right keywords is an important and difficult task.

In this chapter, we propose an intelligent and automatic approach to effective keyword selection. We leverage our knowledge from previous chapters, making use of machine learning to quantify, represent and distinguish semantic information in Tweets and short-texts, to propose two methods for intelligent and automatic keyword selection. The first method takes a heuristic approach [68], while the second method takes an optimization approach, employing evolutionary algorithms [248]. We describe the two methods, and compare and contrast them. For the sake of comparison, we also discuss the manual method of keyword selection, as carried out by humans. We then evaluated the results of all three approaches, discussing our findings. We found that our automatic keyword selection algorithms were able to outperform the manual, human approach.

7.2 Approaches to Intelligent and Automatic Keyword Selection

Before we begin thinking about automatic keyword selection, we must first take a look at how keyword selection normally occurs. While this process might differ based on the purpose of the data collection, it will typically involve some domain knowledge relating to the purpose of the data collection. However, in addition to this, it is often useful to keep in mind that language on Twitter is usually very informal and colloquial. This must also be taken into account when selecting a set of keywords for any purpose. For syndromic surveillance, our goal when selecting keywords for collection was to choose keywords which may be relevant to our particular syndrome of interest. We worked in conjunction with experts from Public Health England (PHE), to create a set of formal terms that may be connected to the specific syndrome under scrutiny. This set of keywords was then further expanded using synonyms from thesauri and the urban dictionary¹. Urban dictionary is a web resource which serves as an encyclopedia of sorts for slangs, so it was used to account for informal language that may occur in Twitter.

Given our understanding of how keyword selection normally occurs, we can begin to build on it in order to arrive at more effective data collection. In this section we propose two approaches to automatic keyword selection. A simple approach would be through trial and error. A set of keywords can be drawn up, used for collection and then assessed. This assessment can be based on the amount of relevant information that was retrieved using the keywords. Depending on the result of the assessment, the terms used are changed or removed. The process can be repeated until a desirable result is obtained from the assessment. While such a process can be automated

¹<https://www.urbandictionary.com>

programmatically, user intervention will be required at the stage of changing or removing keywords. Our first proposed approach works using a similar heuristic method, but sidesteps the need for this user intervention by making use of semantic representations of (key)words which encapsulate the ideas and sentiments behind the terms we use.

Our second approach makes use of evolutionary optimization. Here, the task of keyword selection is modelled as an optimization problem. Each possible set of keywords is seen as a candidate solution and the goal is to find the optimal solution. In this approach, we make use of Particle Swarm Optimization (PSO) which is an evolutionary algorithm based on swarm intelligence put forward by J. Kennedy in 1995 [131]. Loosely speaking, we model each potential set of keywords as a particle in the swarm. Each particle is moved around the search space with some velocity, which is influenced by its known best position, as well as the best positions found by other members of the swarm.

7.2.1 Similarity Heuristic-Based Keyword Selection

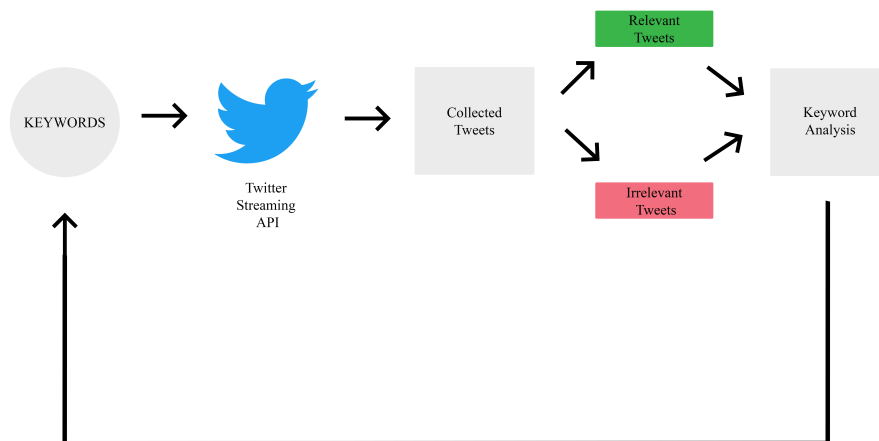


Figure 7.2.1: Flow chart for simple trial and error keyword selection

This method automates the heuristic trial and error technique for keyword selection. The heuristic trial and error technique simply involves producing a set of terms to be used as keywords, employing them and then adjusting the set based on the results obtained. Fig 7.2.1 illustrates this process. Arguably, the important aspect of this approach is how the set of terms is adjusted. It is a trivial matter automating the collection and analysis of its results. As for the matter of adjusting the terms in the keyword set, some more effort is required. The simplest approach to this would be to have human observers examine the results of the collection, together with the keywords used and draw conclusions which can aid their fine-tuning of the keyword set. Unfortunately, this could be time consuming, depending on the number of iterations undertaken. Thankfully, this is not entirely nec-

Algorithm 2 Heuristic-Based Automatic Keyword Selection

```

function KEYWORDSELECTION
   $K \leftarrow \text{INITIALIZEKEYWORDSET}()$ 
   $R\_THRESHOLD \leftarrow C_1$   $\triangleright C_1$  is some numeric constant
   $I\_THRESHOLD \leftarrow C_2$   $\triangleright C_2$  is some numeric constant
  while NOT STOPPING CONDITION  $S$  do
     $T \leftarrow \text{STREAM}(K)$ 
    for all  $k \in K$  do
       $rProp, iProp \leftarrow \text{RELEVANCEANALYSIS}(T, k)$ 
      if  $rProp \geq R\_THRESHOLD$  then
         $Sm \leftarrow \text{FINDSIMILARWORDS}(k)$ 
         $K \leftarrow K + Sm$ 
      else if  $iProp \geq I\_THRESHOLD$  then
         $Sm \leftarrow \text{FINDSIMILARWORDS}(k)$ 
         $K \leftarrow K - k$ 
         $K \leftarrow K - Sm$ 
      end if
    end for
     $S \leftarrow \text{ISSTOPPINGCONDITIONMET}(K)$ 
  end while
end function

```

```

function INITIALIZEKEYWORDSET  $\triangleright$  Returns the initial set of keywords
end function

```

```

function STREAM( $K$ )  $\triangleright$  Run the Tweet streaming using a set of
keywords  $K$ 
end function  $\triangleright$  Returns a list of Tweets collected

```

```

function RELEVANCEANALYSIS( $T, k$ )  $\triangleright T$  : A list of Tweets,  $k$  : A
keyword
 $\triangleright$  Returns the proportions of relevant and irrelevant Tweets in  $T$  for
keyword  $k$ 
end function

```

```

function FINDSIMILARWORDS( $w$ )  $\triangleright w$  : a word
 $\triangleright$  Returns a list of words similar to word  $w$ 
end function

```

```

function ISSTOPPINGCONDITIONMET( $K$ )  $\triangleright K$  : A set of keywords
 $\triangleright$  Returns a boolean answer
end function

```

essary. The analysis of the Tweets obtained from the collection process can be performed using simple statistical analysis. The proportion of relevant and irrelevant Tweets collected can be calculated. Similarly, terms which frequently occur in and associate with relevant and irrelevant Tweets can be uncovered. Once these terms are uncovered, we can carry out any of the following actions to adjust the collection keyword set:

- Terms which appear to associate mostly with irrelevant Tweets can be discarded from the set of keywords, if they were a part of it.
- Terms which are similar to those that associate mostly with irrelevant Tweets can also be discarded if they exist in the collection keyword set.
- Additionally, terms which appear to associate mostly with relevant Tweets can be included in the collection keyword set if they were not previously included.
- Going further, if any terms which appear in the collection keyword set appear to associate strongly with relevant Tweets, other similar terms can be added to the keyword set.

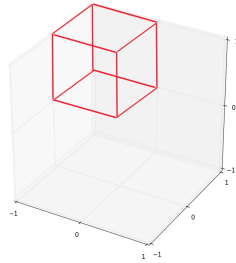
With this in mind, we now consider how to assess the similarities of terms. For this, we encode semantic relationships and meanings using GloVe word embeddings. As previously discussed in chapter 4, such embeddings encode the semantic information of words in continuous distributed vectors, learning this from the contexts within which they are used. We can make use of the cosine distance of these vectors as a measurement of their similarity. With this, we can infer which words are similar to keyword terms that either work well or don't. A more formal explanation of the process is shown in algorithm 2.

Before beginning, a number of variables must be initialized. The first of these is the initial keyword set. This set is seen as a rough guess of the ideal keyword set and serves as the starting point for the algorithm. *R_THRESHOLD* and *I_THRESHOLD* represent the cut-off values for the proportion of relevant and irrelevant Tweets respectively that a keyword brings in for it to either be expanded on in the keyword set or removed from the keyword set. These variables take the values C_1 and C_2 which are floating point numbers. The Stopping condition *S* for the algorithm may differ based on preference or what is desired from the algorithm. Some examples of such condition might involve assessing the total proportion of relevant Tweets collected by the current set of keywords and checking if it is above or below some threshold, or if it has converged or does not appear to be changing significantly. The proposed heuristic-based approach is quite flexible and rather customisable to suit the needs or idiosyncracies of the particular collection task at hand.

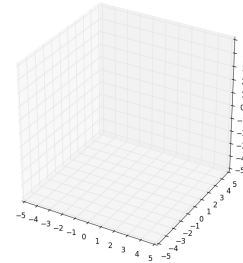
7.2.2 Particle Swarm Optimization-Based Keyword Selection

Evolutionary algorithms and Particle Swarm Optimization (PSO)

Our second proposed approach to keyword selection employs evolutionary computation and algorithms to solve the problem. Evolutionary algorithms are a family of algorithms which are heavily inspired by biology and nature, specifically evolution [171]. The underlying idea behind such algorithms is that given a population of individuals, the environmental pressure causes natural selection and survival of the fittest, thereby improving the fitness of the population. The process of natural selection and survival of the fittest can be seen as a form of optimization. It brings the best individuals in the population to the front, leaving weaker individuals behind. In a computational optimization sense, the population may be represented by the search space and the individuals in said population, represented as potential solutions. The algorithm finds the fittest individual (or solution) in order to solve the optimization problem. Here, the fitness of an individual is modelled by some objective function which we aim to minimize or maximize. The evolutionary algorithm we employ is known as Particle Swarm Optimization (or PSO).



(a) A small example problem space



(b) A larger, more realistic example problem space

Figure 7.2.2: Illustration of different problem search spaces.

PSO is a stochastic population-based algorithm. Unlike other evolutionary algorithms, it does not actually use natural selection; instead, all population members survive from the beginning of a simulation until the end. However, their interactions result in iterative improvement of the quality of solutions over time [292]. With all this in mind, we propose to model the keyword selection task as an optimization problem. Here an individual or particle or solution is a set of keywords. In this problem, the goal is to find the set of keywords that provide the maximum (or minimum) of some objective function. While the obvious solution to such a problem would be simply to check each possible solution in the search space and select the

best, this is not often feasible. Consider the problem space shown in fig 7.2.2a. It is of a low dimensionality and is a very limited space with only 3 possible values for each dimension. The simple approach described earlier would suffice for such a space. However, once in a more realistic space like the one shown in fig 7.2.2b, such an approach is no longer feasible as it would become computationally expensive and time consuming. PSO solves this problem by making use of a set population of particles, where each particle is a potential solution. Fig X shows an illustration of the particles in a problem space. These particles are then set loose to explore the search space in order to find an optimal solution. They tend to swarm and form clusters in optimal regions of the problem space.

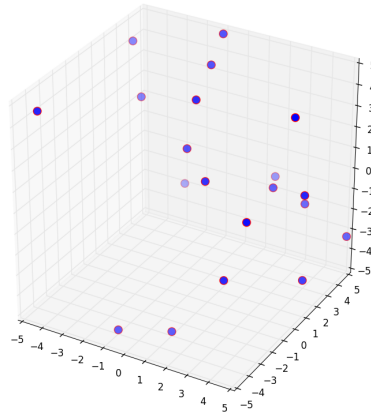


Figure 7.2.3: Illustration of PSO particles in a search space

Applying Particle Swarm Optimization to Keywords

We now look towards how we can model the keyword selection problem using PSO. First we need to produce a larger set of candidate keywords, C , from which our optimal set will be selected. This is done by producing the initial set, I , in the standard way described in section 7.1. Once this has been done, we can build on this initial set by making use of our Glove embeddings to infer words similar to the words in the initial set. That is, for each keyword in the initial set, words similar in meaning to it are added to the initial keyword set to create the candidate keyword set. We make use of the candidate keyword set, C , to collect Tweets, T , for a set period using the streaming API. These Tweets are labelled as *relevant* or *irrelevant* using the classification-based filtering systems discussed in chapter 5. We then take the candidate keyword set, C , and encode each keyword contained within it as a unique integer ranging from 1 to $|C|$. With this, we can now represent a set of keywords as a vector of integers, k , where each integer in the vector maps to a keyword in C . The size of k , denoted as D , must be determined before-hand and equates to the maximum size of the desired final optimal keyword set. Additionally, while values of 1 to $|C|$ represent keywords, a value of zero will be used to represent the absence of a keyword.

With this, when keyword vectors are mapped back to keyword sets, it will be possible to have sets of varying sizes (of up to $|k|$). Having developed a way to represent a set of keywords as a vector, we can also represent a set of keywords as a particle, as a particle is represented by a vector. With this, we can apply PSO to our candidate set, C , to intelligently and automatically select a set of keywords.

We start by randomly initializing a population of particles (i.e. keyword sets) from C . In essence, we create a set number of random vectors of size D , with values ranging from 0 to $|C|$. Each particle possesses a **position**, x and a **velocity**, v , and keeps track of the best position it has found, that is, its “personal best” or ***pbest***. The system keeps track of the “global best” or ***gbest***, which is simply the best position that has ever been found by any particle. The position of the i^{th} particle, $x_i = (x_i^1, x_i^2, x_i^3, \dots, x_i^D)$. The particles are all moved around the search space, with their positions updated based on their velocities, *pbest* values and *gbest*. More formally, after each iteration at time t , the position of the i^{th} particle is updated according to equation 7.2.1

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (7.2.1)$$

The velocity of the particle, v_i^{t+1} , (at time $t + 1$) used to update its position can be computed as shown in equation 7.2.2 where ω is the inertia coefficient, c_1 and c_2 are acceleration coefficients and r_1 and r_2 are random floating point values between 0 and 1.

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pbest - x_i^t) + c_2 r_2 (gbest - x_i^t) \quad (7.2.2)$$

There are three main components to the way the velocity of a particle is updated:

- ***Inertia Component:*** This component is intended to keep the particle moving (or not moving) in the direction it is headed, and is controlled by ω . Lower values of ω will speed up convergence while higher values encourage particle exploration of the search space [260].
- ***Learning Component:*** This component controls the size of the step a particle takes towards its next position in exploring the search space. It is controlled by the coefficient, c_1 [68].
- ***Social Component:*** This component implements swarm mentality, and causes a particle to move towards the best regions the swarm has discovered so far. It is controlled by c_2 [68].

Largely speaking, the particles in the swarm explore the search space based on the *pbests* and *gbest* within the swarm. These values are computed for a particle’s position using the objective function, Z . Our goal is to minimize the value of the Z , which represents the underlying desire of the swarm system. It is the function to be optimized. We make use of

an objective function that is particular to the task of selecting a keyword set. We wish to maximize the number of relevant Tweets (or minimize the number of irrelevant Tweets) collected by a set of keywords. However, we don't wish to achieve this by being too selective and only collecting very few Tweets. Our goal is a combination of the relevancy of the Tweets we collect and the volume of Tweets we collect. Both of these figures are important to us. As such, we developed an objective function that belies this. The objective function, Z , is computed as the mean of two terms, α and β . α is the **irrelevance factor** shown in equation 7.2.3, while β is the **retrieval factor** shown in equation 7.2.4.

$$\alpha = \sum_{i=1}^D \left(\frac{\sum_{j=1}^{|T|} k[i] \in T[j]}{\sum_{j=1}^{|T|} T[j] == \text{irrelevant}} \right) \quad (7.2.3)$$

$$\beta = \sum_{i=1}^D \left(1 - \left(\frac{\sum_{j=1}^{|T|} k[i] \in T[j]}{|T|} \right) \right) \quad (7.2.4)$$

Z is computed as:

$$Z = \frac{\alpha + \beta}{2} \quad (7.2.5)$$

Finally, putting this all together, PSO keyword selection can be carried out by iterating over the following steps:

1. The objective function is computed by each particle for their current position.
2. Each particle updates their *pbest* and the *gbest*.
3. Each particle is moved once their velocity and position are updated, using the *pbest* and *gbest* values computed from the objective function.

The steps are repeated either until the values converge, or a predetermined maximum number of iterations is reached.

7.3 Experiments and Results

We were interested in evaluating whether our proposed approaches solved the task of intelligent and automatic keyword selection and if so, also understanding how well they did so. We implemented and ran Tweet collections for the asthma/difficulty breathing syndrome using both of our keyword selection approaches. We also simultaneously ran a Tweet collection using the typical approach used, which we described in section 7.2, as a baseline for our comparisons.

We undertook two sets of collection periods. The first collection period was a sort of “validation” collection period, inspired by the training/validation/test splits adopted when building data models. This validation collection period was used by our proposed approaches to automatically generate keywords. These generated keywords were subsequently utilized in a second collection period, intended to allow us to measure how well the generated keywords perform. This can be seen as our “test” collection period. Our validation collection period ran for a seven day period from the 24th of May, 2019 till the 1st of July, 2019. Our test collection period ran for a further seven day period from the 1st of July, 2019 till the 8th of July, 2019. Only the similarity heuristic-based approach and the PSO-based approach were involved in the validation collection period, as the baseline approach does not need any data for building. During the test collection period however, all three approaches are involved.

One caveat to consider is that even though the evaluatory Tweet collections were performed simultaneously in parallel, due to the workings of the Twitter API, there is no guarantee that all three collection systems will be exposed to the exact same Tweets at the exact same time. This is because of the fact that the Twitter streaming API only offers a sample of the entire real-time stream, the percentage of which will vary depending on the activity loads at the time. Despite this limitation of the free Twitter API, we should still be able to get some picture of how well our approaches perform. In this section, we describe the experimental setup for each approach, including the baseline standard keyword selection approach. After that, we present and discuss the results we obtained.

7.3.1 Experimental Setup: Baseline Approach

The baseline approach involved working with a group of domain experts to come up with useful keywords and augmenting these keywords with some terms from the Urban Dictionary. We came up with a list of keywords which are included in the appendices. Using these keywords, we ran a Tweet collection during the test collection period, from the 1st of July, 2019, till the 8th of July, 2019. The validation collection period was not used for this part of the experiments as there was no automatic keyword generation, rendering such a period unnecessary.

7.3.2 Experimental Setup: Similarity Heuristic-Based Keyword Selection Approach

This approach involved starting from a minimal set of keyword and expanding and/or reducing this keyword through trial and error and drawing from a pool of terms which the system infers to be similar. This system was applied during the validation collection period starting from the 24th of

Step	Keyword Set
1 st Iteration	<i>asthma</i>
2 nd Iteration	<i>asthma, wheezing, difficulty breathing, trouble breathing, tight chest, inhaler</i>
3 rd Iteration	<i>asthma, wheezing, difficulty breathing, trouble breathing, tight chest, inhaler, hyperventilating, coughing, choking, throat, hurts, breath</i>
4 th Iteration	<i>asthma, wheezing, difficulty breathing, trouble breathing, tight chest, inhaler, hyperventilating, breath</i>

Table 7.1: Keywords at each iteration of the Similarity Heuristic Keyword Selection Process

May, 2019 till the 1st of July 2019. We started with the minimal keyword set consisting solely of the term “*asthma*”. We put in place a stopping condition which terminates the process once a consecutive deterioration in performance between iterations is observed. Each iteration is set to last 24 hours. The system ran, adjusting the keyword set for 4 iterations. Table 7.1 shows the keywords used at each step of the keyword selection process, and the final keywords generated at the end of the 4th iteration. We then applied the final keywords obtained from this method to be used as query inputs in the test collection period.

7.3.3 Experimental Setup: Particle Swarm Optimization-Based Keyword Selection Approach

The standard keyword set used in the baseline approach was used as the seed for creating the candidate set C . For each word in the standard keyword set (which can be found in the appendices), their five most similar words as inferred from our GloVe embeddings were added to the set. This resulted in the large candidate set shown in appendix D. Using the candidate set of keywords, Tweets were collected during the validation collection period. At the end of this period, the PSO-based keyword generation algorithm was applied using the collected Tweets. We set our D , representing the maximum size of a keyword set to be 10. We made use of the PySwarm library of evolutionary algorithms to implement our PSO algorithm. Our

setup had a swarm size of 100. After some experimentation, we set our ω to 0.8., and c_1 and c_2 to 1. This resulted in the following set of keywords being selected as the optimal arrangement: *wheezing, panting, gasping, puffing, couldn't breathe, wheeze, asthma, inhaler, sore eyes*. After obtaining the automatically selected keywords, we applied them during the test collection period, using them as query inputs.

7.4 Results

We utilized the two sets of keywords we obtained from our keyword selection algorithms as query inputs for Tweet collection. We also utilized the keywords obtained using the standard baseline approach. We applied the three distinct sets of keywords in parallel during our test collection period - 1st of July, 2019, till the 8th of July, 2019. We then analyzed the Tweets collected by each set of keywords in order to understand how useful each keyword set was. We assessed the keyword sets based on their information retrieval ability. A lot of the traditional information retrieval metrics do not translate well, or cannot be calculated for our problem. For example, recall, which measures the fraction of relevant documents retrieved cannot be calculated because we have no way of knowing the total amount of relevant Tweets out there. Because of this, we made use of a combination of traditional metrics and developed problem-specific metrics. These metrics are ***precision*** and ***reach***.

Precision is a popular information retrieval metric which represents the proportion of retrieved documents which are relevant. In such an information retrieval context, precision is calculated as:

$$precision = \frac{|RelevantTweets| \cap |CollectedTweets|}{|CollectedTweets|} \quad (7.4.1)$$

In our scenario, precision measures the proportion of the collected Tweets which are relevant. When calculating the precision values for each keyword approach, we computed the precision over a random sample of the retrieved Tweets. We took random 2000-large samples of the Tweets collected using each keyword selection approach and computed the precision from this sample. We did this because we wanted to manually label and count the number of relevant Tweets, instead of relying on one of our trained classifiers which are not perfect. Doing so allowed us to get an accurate and exact value for the number of relevant Tweets, and would not be feasible with the complete set of collected Tweets which are very large and would be incredibly time-consuming to manually label.

Reach is a metric we developed to help us capture the ability of a set of keywords to retrieve as many Tweets as possible, relevant or not. This is important because while it is useful to collect relevant Tweets, if we only

Keyword Selection Approach	Precision	Reach	TRP
Baseline Human Approach	0.23	0.75	0.35
Similarity Heuristic Approach	0.40	0.64	0.49
PSO Approach	0.48	0.65	0.55

Table 7.2: Performances of different approaches to keyword selection

observe a small amount of Tweets, we cannot create a useful signal which is appropriately representative of the activity related to the syndrome of interest. As such, reach measures the quantity of Tweets a set of keywords is able to collect. This could be computed simply as the proportion of the general Tweet stream that is collected using a set of keywords. However, the inner workings of the Twitter API is unknown to us. To overcome any bias introduced by the API and any rate limits it may impose, we calculate the *reach* of a set of keywords in relation to the simplest singular keyword possible. This can be formally represented as shown below:

$$reach = \frac{|CollectedTweets|_{\hat{k}} - |CollectedTweets|_K}{|CollectedTweets|_K} \quad (7.4.2)$$

\hat{k} represents some arbitrary single unit keyword which is a simple and straightforward keyword. For example, in our scenario of *asthma/difficulty breathing* surveillance, we make use of the keyword “asthma” as \hat{k} .

Finally, we combined the precision and reach metrics into one metric by taking their harmonic mean, similar to the *F*-measure. We term this combined metric, the ***Tweet Retrieval Power (TRP)***.

$$TRP = 2 \frac{precision \times reach}{precision + reach} \quad (7.4.3)$$

The TRP weights precision and reach evenly but similarly to the *F*-measure, it is possible to calculate variations of the TRP score which place different weights on precision and reach as below:

$$TRP_{\beta} = (1 + \beta^2) \frac{precision \times reach}{\beta^2(precision + reach)} \quad (7.4.4)$$

where TRP_{β} measures the Tweet retrieval ability when β times as much importance is placed on reach than precision.

Table 7.2 shows the results observed at the end of our analysis. We found the PSO approach to have the best Tweet Retrieval Power. Both the PSO approach and the similarity heuristic approach yielded fair improvements in

precision and reach over the baseline human approach. These approaches also resulted in a decrease in reach however. The PSO approach only gives a modest improvement in reach over the similarity heuristic approach. In fact this improvement is very small and there could be some question over its statistical significance. Unfortunately, computing the reach is a time consuming process, as one calculation necessitates a collection endeavor for a period of one week. At this time, we cannot feasibly repeat this process sufficient times to test for statistical significance. On the precision side of things, the PSO approach appears to produce clear improvements and be the best bet for collecting relevant Tweets. All in all, the PSO approach yields the highest TRP and appears to possess a reasonable balance of precision and reach.

7.5 Discussion

In this section, we investigated methods of intelligently and automatically selecting keywords for use in collecting Tweets. We proposed two techniques for accomplishing this. The first approach was a heuristic method which made use of similarity to automatically expand and reduce a keyword set, in an iterative manner. The second approach was an evolutionary algorithm inspired method which modelled the keyword selection task as an optimization problem. It made use of Particle Swarm Optimization (PSO) to determine the optimal set of keywords. We implemented and applied both approaches to the task of collecting Tweets for the surveillance of the *asthma/difficulty breathing* syndrome. For the sake of comparison, we also carried out a Tweet collection with keywords selected using a manual, human approach. We then evaluated the results of all three approaches, making comparisons between them.

We found that the PSO-based method performed the best, outperforming the manual, human approach by a fair margin. The similarity heuristic method also performed better than the baseline human approach but was not as strong as the PSO method. While we observed a fair increase in relevance (precision) using our automatic keyword selection algorithms, we saw the opposite when looking at the reach metric. The baseline human approach to curating keywords seemed to have the most reach. Despite this, the boost in precision offered by the automatic keyword selection algorithms meant that they outperformed the baseline approach, with both algorithms yielding higher TRP values. However, it is also important to remember that while we tried to keep things constant in our experiments, applying each keyword selection approach in parallel during the same periods, we cannot guarantee that they were exposed to the same environments and Tweets as that is an issue dependent on the Twitter API. Studies have estimated that using the Twitter streaming API, users can expect to receive anywhere from 1% of the tweets to 40% of tweets available in real-time, depending on the amount of activity at the time [183].

While we have introduced these techniques for the intelligent and automatic selection of keywords and used them for surveilling the syndrome of *asthma/difficulty breathing*, this does not mean it does not generalize to other tasks. These techniques cannot only be applied for the purposes of surveilling other syndromes on Twitter, but also for any Tweet collection exercise, regardless of the purpose of said exercise. This is due to the fact that these techniques aim to maximize the relevance of collected Tweets to some query, together with the volume of Tweets collected. As long as there exists some defined query, the notion of “relevance” for its results must also exist. Because these are the main ideas behind our proposed approaches, they can be very easily adapted to any other problem and generalize very well.

While both the similarity heuristic approach and PSO approach work better than the manual approach, we have established that the PSO method yield the best results. In addition, the PSO approach is a lot less time consuming to run. This is because the similarity heuristic method involves a series of automated trial and error iterations which allow it to make adjustments to the keyword set. To adequately evaluate a trial, a fair amount of data must be available for it. This means that a trial (or single iteration), would typically have to last 1 day. Naturally, multiple iterations means the process of selecting a set of keywords can take multiple days. This is not the case for the PSO method. Recall that for the PSO method, the data used to evaluate the objective function is all collected in advance. As such, the time taken for each iteration is determined by how many particles the swarm contains and the dimension of each particle, as well as the computational resources available. Even with a mid-tier computer, a single iteration could never take longer than an hour in the absolute worst case. As such the PSO method is not only superior in terms of the quality of the keywords produced, but also in terms of the amount of time taken to produce said keywords.

Chapter 8

Conclusions of the Thesis

This chapter summarizes the contributions of the research carried out as part of this thesis. It begins with a chapter-by-chapter synopsis of the contents of this work. We then discuss the general outcomes of the research, looking back in hindsight to our aims and objectives, while also highlighting its impact and contributions. Finally, we finish by proposing directions for further research that may be interesting to explore in the future.

8.1 Summary of Thesis

The main goal of this project was to establish whether social media data, and specifically, Twitter data can be used in the context of syndromic surveillance in order to generate or contribute to alarms for a specific (non-infectious) syndrome - asthma/difficulty breathing - in the UK. After applying various preprocessing operations to the Twitter data, we needed to filter relevant Tweets as even though Tweets contained telling keywords such as “asthma”, they were not always related to individuals who were symptomatic or concerned, and were sometimes merely reference or passing mentions. We found that the quality of the Twitter syndromic surveillance system was improved with relevance filtering, with an attentive bi-directional LSTM performing best at the task. The syndromic Twitter signal obtained from the attentive bi-directional LSTM had the strongest correlation with the ground truth. We also discovered that that Twitter has some potential for detecting public health incidents before traditional surveillance systems. We performed some lag analysis between our generated Twitter signals and ground-truth syndromic surveillance signals. The results of this analysis suggest that we can use Twitter to catch wind of

trends in public health before they reach formal reporting routes. Lastly, we proposed that this, coupled with an evolutionary algorithm approach to automatic keyword selection for streaming Tweets in, resulted in a performant syndromic surveillance system. Figure 8.1.1 shows the signal obtained from our proposed methodology.

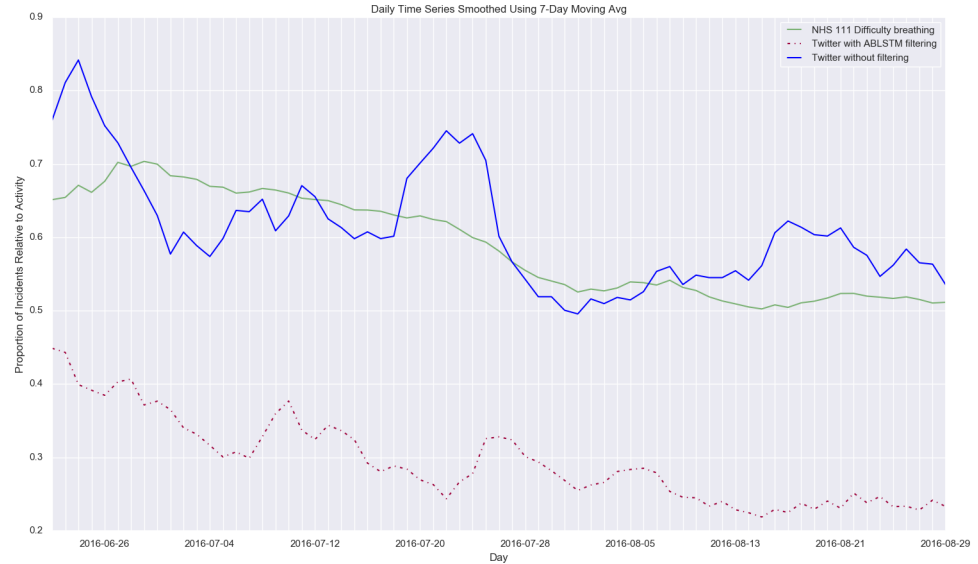


Figure 8.1.1: Time series plot showing generated Twitter syndromic surveillance signal with a ground truth signal for comparison

In **Chapter 1**, we introduced our project, highlighting our aims and objectives and describing the motivations to undergo the project. Here, we set the stage for the rest of the thesis, providing an outline of this script and finally briefly highlighting the research outputs and contributions of this project to the wider academic community.

Chapter 2 presented a technical and methodological background, familiarizing the reader with the relevant fields of pattern recognition, natural language processing and statistical machine learning, and summarized popular algorithms, techniques and ideas in these fields. This chapter introduced theoretical notions which aid the reader in understanding the thesis moving forward.

In **Chapter 3**, we carried out a comprehensive review of the literature surrounding the use of Twitter data for public health purposes, including but not limited to syndromic surveillance. Through this review, we gain a clear picture of the state of the literature, understanding research attitudes and directions. In doing so, we are able to identify state-of-the-art applications and algorithms, as well as gaps in the field, which helped motivate

and guide the direction of this project.

Chapter 4 gave a characterization of Twitter describing its main attributes and offerings. In this chapter, we explain in detail how we collect, preprocess and store data from Twitter. We also described the data collected from Twitter. In the second part of this chapter, we proposed and described our approaches to extracting meaningful features from the collected Twitter data and producing useful feature representations of it. We introduced novel hand-crafted features, some of which capitalize on emoji content in their construction. Embedding-based tweet feature representations were also explored in this chapter.

In **Chapter 5**, we identified the steps vital to syndromic surveillance using Twitter data. In doing so we recognized the need for further filtering of Tweets based on their relevance to a syndrome of interest, as well as its exigency to our syndromic surveillance efforts. In order to build a reliable syndromic surveillance signal, we need to extract only Tweets expressing discomfort and/or concern related to a syndrome of interest, and reflecting current events. In this chapter, we proposed various methodologies for effectively achieving this. We focused on algorithms based on semi-supervised learning ideas, as very little research has been carried out in the area and such techniques allowed us to minimize our labelling efforts, make use of our vast amounts of labelled data. We described an iterative labelling methodology for Tweet relevance filtering. We also looked towards (semi-supervised) algorithms based on the prevalent and powerful field of deep learning. In doing so, we also experimented with a novel classification algorithm based on neural language models termed the Generative Classification Network (GCN), comparing it to popular deep learning classification algorithms. Finally, we also proposed a novel attention based bi-directional Recurrent Neural Network.

In **Chapter 6**, we described the experiments we carried out to evaluate our proposed methodologies and approaches to Tweet relevance filtering, along with our results. We also described our experimental evaluations of our various proposed feature extraction and representation approaches. Emojis were found to make informative and useful features. On the side of relevance filtering, semi-supervised iterative labelling was observed to outperform popular fully-supervised algorithms at the task of Tweet relevance filtering. Our experimental GCN deep classification algorithm was outperformed by RNNs at the task of relevance filtering. However, it was also able to give better results than CNNs at the same task. We found the attentive bi-directional RNN to perform the best at the task of relevance filtering. In the second part of this chapter, we directly evaluated and compared the signals generated by our proposed methodologies. The attentive bi-directional RNN was found to produce the best signals, giving us a strong and statistically significant correlation with real public health

data ($r = 0.830$), confirming the utility for Twitter as a data source for syndromic surveillance in the UK.

Lastly, **Chapter 7** narrated a novel study into the improvement of the Twitter syndromic surveillance pipeline by intelligently and automatically selecting the optimal keywords to collect as relevant, and as many Tweets as possible, in order produce a better signal. We proposed and compared two algorithms which make use of deep learning to this effect: (a) an iterative heuristic approach and (b) an evolutionary computing approach. We found that the second approach, which modelled the task of automatic keyword selection as an optimization problem, performed better. This approach was also found to outperform the baseline human approach, resulting in a better quality of collected Tweets, as well as requiring less manual effort and time.

8.2 Research Conclusions

This research project was carried out with the goal of establishing Twitter's utility for syndromic surveillance and the development of techniques for such. In doing so, a number of questions were raised which we aimed to answer through our research. We summarize the results of this project by discussing the answers we have learned from it.

Q1: Is there useful, extratable information in the large amounts of Twitter data available?

Through our comprehensive scoping review of the use of Twitter data for public health purposes, we obtained some insight into the value of Twitter as a data source, as well as the quality of information present in Twitter data. We confirmed our hypothesis that the enormous volumes of data contained some meaningful, useable information. We learned that Twitter data can be effectively employed for multitudinous purposes ranging from surveillance to pharmacovigilance to the capturing of environmental and social issues. With all of this in mind, we moved forward in our project to understand how we can employ Twitter data for syndromic surveillance in the UK.

Q2: How can useful information effectively and efficiently be extracted from Twitter?

We sought to answer this question specifically for the context of syndromic surveillance in the UK. This was the heart of the research carried out in this project. We investigated a number

of supervised, semi-supervised and unsupervised algorithms for extracting information from Twitter. In doing so, we also developed our own algorithms to this effect. We found that emojis proved helpful in constructing feature to better understand and classify Tweets. We observed a distinct absence of the use of semi-supervised learning being applied to extract information from Twitter within the relevant literature. However, due to the advantages which semi-supervised learning has to offer, we investigated its utility for our goals. Semi-supervised iterative labelling was found to outperform popular fully-supervised algorithms at extracting information from Twitter for syndromic surveillance. In addition, we successfully applied deep learning algorithms to the understanding and classification Twitter data. Our proposed GCN algorithm proved somewhat useful, as it was observed to outperform the popular CNN deep learning classification algorithm. However, it was itself outperformed by RNNs. Learning from this, we developed an attention-based bi-directional RNN which not only allowed us to accurately and efficiently identify relevant Tweets for syndromic surveillance, but also allowed us to automatically identify and extract keywords in collected Tweets. Lastly, we also developed algorithms for the intelligent and automatic selection of keywords to improve the Twitter data collection. We learned that this was possible through a combination of deep learning and evolutionary computing, specifically Particle Swarm Optimization (PSO). We found that we were able to come up with better keywords and collect more relevant Tweets.

Q3: Does the information extracted from Twitter mirror the real-world such that it is a reasonable data source for syndromic surveillance?

While we found that we were able to effectively and efficiently extract meaningful information surrounding syndromic surveillance from collected Twitter data, this alone did not confirm its potential for syndromic surveillance. We were able to automatically identify symptomatic and relevant Tweets, but this did not answer the question of whether Twitter could be a reasonable data source for syndromic surveillance. The results of our experiments using real-world syndromic surveillance data provided by Public Health England (PHE) as ground truth helped us answer this question. We made use of our developed algorithms to extract a signal from Twitter data. We then also extracted a signal from the ground truth data. We compared these signals and found a strong and statistically significant correlation between them. As such, we were able to confirm that Twitter

sufficiently mirrors the real-world to the point that it could be a potential data source for syndromic surveillance.

8.3 Research Novelty and Contributions

There were a number of novel aspects and contributions generated through the research carried out as part of this project. Some of these novelties have been published in scientific journals and conference proceedings, while others are either under consideration for publication or in the process of being submitted for publication at the time of writing this thesis. The novelty and contributions of our research can be summarized as follows:

- We carried out a comprehensive and methodical scoping review to map the field of Twitter mining for public health purposes from January 2009 till March 2019. This study identified and described the various ways in which Twitter has been mined for health purposes over a ten year period, as well as algorithms, ideas and approaches implemented. As such, it is of use to any researchers interested in obtaining some insight into the field or some direction or clues as to what techniques may be worth applying. It is currently under review for publication in the *European Journal of Public Health*.
- We introduced and verified the efficacy of semi-supervised learning within the context of syndromic surveillance. This contribution was published as a journal article in *PLOS One*. In this study, we showed that the collected and unlabelled data did not have to go to waste and could in fact, improve results with minimal labelling efforts. According to our scoping review, other similar studies seem to make use of an average of 10,000 labelled Tweets. Despite the fact that we only labelled 3500 Tweets, we obtained competitive results and outperformed a number of fully supervised algorithms.
- As part of the above published study, we also highlighted the potential and discriminatory power of emojis in text classification problems, together with capable features for Tweet classification within the context of syndromic surveillance.
- We also developed the Generative Classification Network (GCN), a novel and experimental classification algorithm based on deep generative neural network models. We carried out a study comparing the GCN to other deep learning approaches to relevance filtering and Tweet classification which was published as part of the 2019 *International Conference on Pattern Recognition Applications and Methods*. The GCN was outperformed by RNNs at Tweet classification, but performed better than CNNs at the same task. While this algorithm does

not obtain better results than the current state-of-the-art, it still outperforms some popular and powerful models. Also, the GCN relies on language models as part of its classification process. Language models are usually built using large amounts of unlabelled data. However, because the GCN uses language models in a supervised way, it needs labelled data to construct its language models. As such, the GCN is a very data-hungry algorithm and would require large amounts of data to perform at its best. While this is of course, a disadvantage, we believe the GCN could still be of some merit in some scenarios.

- We experimented with an attentive bi-directional RNN architecture and applied it to the task of syndromic surveillance. This architecture was found to be effective at the task of relevance filtering and also produced reliable syndromic surveillance signals. A conference paper detailing this architecture and our applications was published as part of the conference proceedings of the 2019 *International Work-Conference on Artificial Neural Networks*. The paper was well received and we have been invited to submit an extended version for consideration in a special issue of *PLOS One*.
- We also introduced a general framework for the automatic and optimal selection of keywords for Twitter data collection based on evolutionary algorithms and deep learning. This framework minimised manual efforts and human intervention, while simultaneously maximizing the quality and quantity of Tweets collected using the Twitter API.

8.4 Limitations and Directions for Future Work

The results for syndromic surveillance using Twitter data were promising. However, this work could still be extended in a number of different directions. Firstly, the location filtering aspect of our methodology only relied on the Twitter API. The avenues for location filtering provided by the Twitter API are not entirely reliable. Further research could be carried out looking into more sophisticated ways of inferring the geographical origin of a Tweet. This problem was found to be an active research area through our scoping review. However, this field of Twitter mining has seen the least amount of activity when looking at Twitter mining for public health. More attention should be paid to tackling this problem in future work.

Trying to improve the core theories of the proposed methodologies is always a viable direction for future research. Some work could be done creating more complex features, as well as more powerful neural network architectures to aid the syndromic surveillance effort. Additionally, including further syndromic case studies to the experiments may help us understand which key properties our learning functions require. While a lot of research

has previously been carried out applying Twitter to infectious diseases, and this project tries to look towards non-infectious diseases, we only look at *asthma/difficulty breathing*. Other non-infectious diseases could be investigated. Furthermore, the utility of Twitter data for the syndromic surveillance of other kinds of diseases and public health threats could also be investigated. Examples of such kinds of threats are sexually transmitted diseases, mental health disorders and riot or terrorist activity.

Twitter is only one of a number of different and unique social media platforms. Another avenue for future research would be to examine the utility of other social media platforms for syndromic surveillance. The ways through which users interact with different social media platforms can vary. For example the main forms of media on Snapchat are images and videos. Facebook allows long form texts and videos, and images. On Instagram, users make use of images and short videos. The methodologies involved in text-centric Twitter mining will not readily lend themselves to Instagram mining for example. Research into using some of these social media platforms will require new methodologies created to suit the nature of the content on their platforms.

Such future research could also involve the use of multimodal data sources. While Twitter does offer the opportunity for multimodal approaches to data mining, such approaches would not be particularly useful within the context of syndromic surveillance. Most Tweets do not contain images. Additionally, people suffering from health symptoms, for example asthma, do not tend to post pictures to go with their symptom complaints. On the other hand, posts on Instagram take the form of captioned images. Algorithms which consider both the text and image data could be investigated to handle such a platform. Video classification, which poses an important and interesting challenge, could also be investigated for mining video blogging platforms such as YouTube. Furthermore, signals could be collected from multiple social media platforms simultaneously and combined in a meaningful way to produce more complex and/or detailed signals for syndromic surveillance.

As a final but important note, we would like to highlight the need for the ethical issues and implications surrounding social media surveillance to be seriously considered while pushing for new developments. There is cause for concern over the privacy rights of social media users from whom the data is generated. Developments in Twitter mining, albeit for syndromic surveillance and public health purposes, could indirectly compromise the safety of Twitter users. Future work in the area should include social scientists and legislators before putting such surveillance systems into production, in order to ensure that society's freedom of speech and right to privacy is not infringed upon.

Bibliography

- [1] Nahla B Abdel-Hamid, Sally ElGhamrawy, Ali El Desouky, and Hesham Arafat. A dynamic spark-based classification framework for imbalanced big data. *Journal of Grid Computing*, 16(4):607–626, 2018.
- [2] Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. Predicting flu trends using twitter data. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, apr 2011.
- [3] Cosme Adrover, Todd Bodnar, Zhuojie Huang, Amalio Telenti, and Marcel Salathé. Identifying adverse effects of HIV drug treatment and associated sentiments using twitter. *JMIR Public Health and Surveillance*, 1(2):e7, jul 2015.
- [4] Oluwaseun Ajao, Jun Hong, and Weiru Liu. A survey of location inference techniques on twitter. *Journal of Information Science*, 41(6):855–864, 2015.
- [5] Noraida Haji Ali and Noor Syakirah Ibrahim. Porter stemming algorithm for semantic checking. In *Proceedings of 16th International Conference on Computer and Information Technology*, pages 253–258, 2012.
- [6] Chris Allen, Ming-Hsiang Tsou, Anoshe Aslam, Anna Nagel, and Jean-Mark Gawron. Applying GIS and machine learning methods to twitter data for multiscale surveillance of influenza. *PLOS ONE*, 11(7):e0157734, jul 2016.
- [7] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
- [8] Periklis Andritsos, Panayiotis Tsaparas, Renée J Miller, and Kenneth C Sevcik. Limbo: Scalable clustering of categorical data. In

- International Conference on Extending Database Technology*, pages 123–146. Springer, 2004.
- [9] Yin Aphinyanaphongs, Armine Lulejian, Duncan Penfold Brown, Richard Bonneau, and Paul Krebs. Text classification for automatic detection of e-cigarette use and use for smoking cessation from twitter: a feasibility pilot. In *Biocomputing 2016: Proceedings of the Pacific Symposium*, pages 480–491. World Scientific, 2016.
- [10] MAC Apple. Os x spotlight, 2012.
- [11] Hilary Arksey and Lisa O’Malley. Scoping studies: towards a methodological framework. *International journal of social research methodology*, 8(1):19–32, 2005.
- [12] Masayuki Asahara and Yuji Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology- Volume 1*, pages 8–15. Association for Computational Linguistics, 2003.
- [13] R Auxilia and Meera Gandhi. Earthquake reporting system development by tweet analysis with approach earthquake alarm systems. *RESEARCH JOURNAL OF PHARMACEUTICAL BIOLOGICAL AND CHEMICAL SCIENCES*, 7(3):501–506, 2016.
- [14] Vimala Balakrishnan and Ethel Lloyd-Yemoh. Stemming and lemmatization: a comparison of retrieval performances. *Lecture Notes on Software Engineering*, 2(3):262, 2014.
- [15] Florian Beil, Martin Ester, and Xiaowei Xu. Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 436–442. ACM, 2002.
- [16] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.
- [17] Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai, 2007.
- [18] Shane Bergsma, Matt Post, and David Yarowsky. Stylometric analysis of scientific articles. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–337. Association for Computational Linguistics, 2012.

- [19] Jiang Bian, Umit Topaloglu, and Fan Yu. Towards large-scale twitter mining for drug-related adverse events. In *Proceedings of the 2012 international workshop on Smart health and wellbeing - SHB '12*. ACM Press, 2012.
- [20] Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201. Association for Computational Linguistics, 1997.
- [21] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [22] blog.btrax.com. How americans and the japanese use emoji differently. <https://blog.btrax.com/how-americans-and-the-japanese-use-emoji-differently/>, 2015.
- [23] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.
- [24] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.
- [25] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.
- [26] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Nyu: Description of the mene named entity system as used in muc-7. In *In Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Citeseer, 1998.
- [27] Thorsten Brants. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics, 2000.
- [28] Patrick Breen, Jane Kelly, Timothy Heckman, and Shannon Quinn. Mining pre-exposure prophylaxis trends in social media. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, oct 2016.
- [29] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational linguistics*, 21(4):543–565, 1995.

- [30] David A. Broniatowski, Michael J. Paul, and Mark Dredze. National and local influenza surveillance through twitter: An analysis of the 2012-2013 influenza epidemic. *PLoS ONE*, 8(12):e83672, dec 2013.
- [31] David Andre Broniatowski, Mark Dredze, Michael J Paul, and Andrea Dugas. Using social media to perform local influenza surveillance in an inner-city hospital: a retrospective observational study. *JMIR public health and surveillance*, 1(1), 2015.
- [32] Sebastian Bruch, Xuanhui Wang, Mike Bendersky, and Marc Najork. An analysis of the softmax cross entropy loss for learning-to-rank with binary relevance. 2019.
- [33] Wray Buntine, Jaakko Lofstrom, Jukka Perkio, Sami Perttu, Vladimir Poroshin, Tomi Silander, Henry Tirri, Antti Tuominen, and Ville Tuulos. A scalable topic-based open source search engine. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 228–234. IEEE Computer Society, 2004.
- [34] Kenny Byrd, Alisher Mansurov, and Olga Baysal. Mining twitter data for influenza detection and surveillance. In *Proceedings of the International Workshop on Software Engineering in Healthcare Systems*, pages 43–49. ACM, 2016.
- [35] Claire Cardie, Janyce Wiebe, Theresa Wilson, and Diane J Litman. Combining low-level and summary representations of opinions for multi-perspective question answering. In *New directions in question answering*, pages 20–27, 2003.
- [36] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. Semi-supervised learning, vol. 2. *Cambridge: MIT Press. Cortes, C., & Mohri, M.(2014). Domain adaptation and sample bias correction theory and algorithm for regression. Theoretical Computer Science*, 519:103126, 2006.
- [37] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in neural information processing systems*, pages 601–608, 2003.
- [38] Lauren E Charles-Smith, Tera L Reynolds, Mark A Cameron, Mike Conway, Eric HY Lau, Jennifer M Olsen, Julie A Pavlin, Mika Shigematsu, Laura C Streichert, Katie J Suda, et al. Using social media for actionable disease surveillance and outbreak management: a systematic literature review. *PloS one*, 10(10):e0139701, 2015.
- [39] Michael Chary, Nicholas Genes, Christophe Giraud-Carrier, Carl Hanson, Lewis S. Nelson, and Alex F. Manini. Epidemiology from tweets: Estimating misuse of prescription opioids in the USA from social media. *Journal of Medical Toxicology*, 13(4):278–286, aug 2017.

- [40] Liangzhe Chen, K. S. M. Tozammel Hossain, Patrick Butler, Naren Ramakrishnan, and B. Aditya Prakash. Syndromic surveillance of flu on twitter using weakly supervised temporal topic models. *Data Mining and Knowledge Discovery*, 30(3):681–710, sep 2015.
- [41] Liangzhe Chen, K.S.M. Tozammel Hossain, Patrick Butler, Naren Ramakrishnan, and B. Aditya Prakash. Flu gone viral: Syndromic surveillance of flu on twitter using temporal topic models. In *2014 IEEE International Conference on Data Mining*. IEEE, dec 2014.
- [42] Liangzhe Chen, K. S. M. Tozammel Hossain, Patrick Butler, Naren Ramakrishnan, and B. Aditya Prakash. Syndromic surveillance of flu on twitter using weakly supervised temporal topic models. *Data Mining and Knowledge Discovery*, 30(3):681–710, 2016.
- [43] Yi-Song Chen, Guo-Ping Wang, and Shi-Hai Dong. A progressive transductive inference algorithm based on support vector machine. *Journal of Software*, 14(3):451–460, 2003.
- [44] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [45] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [46] D Paice Chris et al. Another stemmer. In *ACM SIGIR Forum*, volume 24, pages 56–61, 1990.
- [47] Grzegorz Chrupała. Simple data-driven contextsensitive lemmatization. *Procesamiento del Lenguaje Natural*, 37:121–127, 2006.
- [48] William J Clancey. *Classification problem solving*.
- [49] J. Clement. Number of Twitter users worldwide from 2014 to 2020. <https://www.statista.com/statistics/303681/twitter-users-worldwide/>, 2019. Accessed: 2019-07-30.
- [50] Ira Cohen and Thomas S Huang. Semisupervised learning of classifiers with application to human-computer interaction. *University of Illinois at Urbana-Champaign, Champaign, IL*, 2003.
- [51] Kevyn Collins-Thompson. Computational assessment of text readability: A survey of past, present, and future research. *International Journal of Applied Linguistics*, 165(2):97–135, 2014.

- [52] Michael D Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. In *Fifth international AAAI conference on weblogs and social media*, 2011.
- [53] Paige Coope. Number of Twitter users worldwide from 2014 to 2020. <https://blog.hootsuite.com/twitter-statistics/>, 2019. Accessed: 2010-07-30.
- [54] W Bruce Croft. *Organizing and searching large files of document descriptions*. PhD thesis, University of Cambridge, 1978.
- [55] Aron Culotta. Towards detecting influenza epidemics by analyzing twitter messages. In *Proceedings of the first workshop on social media analytics*, pages 115–122. ACM, 2010.
- [56] Aron Culotta. Estimating county health statistics with twitter. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. ACM Press, 2014.
- [57] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 133–140. Association for Computational Linguistics, 1992.
- [58] Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329. ACM, 1992.
- [59] Xiangfeng Dai and Marwan Bikdash. Hybrid classification for tweets related to infection with influenza. In *SoutheastCon 2015*. IEEE, apr 2015.
- [60] Xiangfeng Dai and Marwan Bikdash. Distance-based outliers method for detecting disease outbreaks using social media. In *SoutheastCon 2016*. IEEE, mar 2016.
- [61] Xiangfeng Dai, Marwan Bikdash, and Bradley Meyer. From social media to public health surveillance: Word embedding based clustering method for twitter classification. In *SoutheastCon 2017*. IEEE, mar 2017.
- [62] Sanjiv Das and Mike Chen. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)*, volume 35, page 43. Bangkok, Thailand, 2001.

- [63] Kushal Dave, Steve Lawrence, and David M Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.
- [64] John Dawson. Suffix removal and word conflation. *ALLC bulletin*, 2(3):33–46, 1974.
- [65] Soham De, Anirbit Mukherjee, and Enayat Ullah. Convergence guarantees for rmsprop and adam in non-convex optimization and an empirical comparison to nesterov acceleration. 2018.
- [66] Cecilia de Almeida Marques-Toledo, Carolin Marlen Degener, Livia Vinhal, Giovanini Coelho, Wagner Meira, Claudia Torres Codeço, and Mauro Martins Teixeira. Dengue prediction by the web: Tweets are a useful tool for estimating and forecasting dengue at country and city level. *PLOS Neglected Tropical Diseases*, 11(7):e0005729, jul 2017.
- [67] Ed De Quincey and Patty Kostkova. Early warning and outbreak detection using social networking websites: The potential of twitter. In *International Conference on Electronic Healthcare*, pages 21–24. Springer, 2009.
- [68] Kalyanmoy Deb and Nikhil Padhye. Improving a particle swarm optimization algorithm using an evolutionary algorithm framework. *Kan-GAL Report*, 2010:003, 2010.
- [69] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391, 1990.
- [70] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [71] Mohit Deshpande. Recurrent neural networks for language modeling. <https://pythonmachinelearning.pro/recurrent-neural-networks-for-language-modeling/>.
- [72] Jeff Desjardins. How much data is generated each day? . <https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/>, 2019. Accessed: 2019-07-30.
- [73] Ernesto Diaz-Aviles and Avaré Stewart. Tracking twitter for epidemic intelligence. In *Proceedings of the 3rd Annual ACM Web Science Conference on - WebSci '12*. ACM Press, 2012.

- [74] Inoshika Dilrukshi, Kasun De Zoysa, and Amitha Caldera. Twitter news classification using SVM. In *Computer Science & Education (ICCSE), 2013 8th International Conference on*, pages 287–291. IEEE, 2013.
- [75] Shifei Ding, Zhibin Zhu, and Xiekai Zhang. An overview on semi-supervised support vector machine. *Neural Computing and Applications*, 28(5):969–978, 2017.
- [76] Vanessa Doctor. What is a retweet? <https://www.hashtags.org/featured/what-is-a-retweet/>, 2012.
- [77] Alex J Elliot, Sue Smith, Alec Dobney, John Thornes, Gillian E Smith, and Sotiris Vardoulakis. Monitoring the effect of air pollution episodes on health care consultations and ambulance call-outs in england during march/april 2014: A retrospective observational analysis. *Environmental pollution*, 214:903–911, 2016.
- [78] Alvaro Esperanca, Zina Ben Miled, and Malika Mahoui. Social media sensing framework for population health. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, jan 2019.
- [79] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.
- [80] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.
- [81] Andrea Gesmundo and Tanja Samardžić. Lemmatisation as a tagging task. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 368–372. Association for Computational Linguistics, 2012.
- [82] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Clustering categorical data: An approach based on dynamical systems. *Databases*, 1:75, 1998.
- [83] Antonio A. Ginart, Sanmay Das, Jenine K. Harris, Roger Wong, Hao Yan, Melissa Krauss, and Patricia A. Cavazos-Rehg. Drugs or dancing? using real-time machine learning to classify streamed “dabbing” homograph tweets. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, oct 2016.
- [84] Jeremy Ginsberg, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012, 2009.

- [85] Jeremy Ginsberg, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012, 2009.
- [86] Janaína Gomide, Adriano Veloso, Wagner Meira, Virgílio Almeida, Fabrício Benevenuto, Fernanda Ferraz, and Mauro Teixeira. Dengue surveillance based on a computational model of spatio-temporal locality of twitter. In *Proceedings of the 3rd International Web Science Conference on - WebSci '11*. ACM Press, 2011.
- [87] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.
- [88] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Coling*, volume 96, pages 466–471, 1996.
- [89] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. In *ACM Sigmod Record*, volume 27, pages 73–84. ACM, 1998.
- [90] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366, 2000.
- [91] Shashank Gupta, Sachin Pawar, Nitin Ramrakhiyani, Girish Keshav Palshikar, and Vasudeva Varma. Semi-supervised recurrent neural network for adverse drug reaction mention extraction. *BMC Bioinformatics*, 19(S8), jun 2018.
- [92] A. A. Hamed, R. Roose, M. Branicki, and A. Rubin. T-recs: Time-aware twitter-based drug recommender system. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, aug 2012.
- [93] CL Hankin, O Serban, N Thapen, B Maginnis, and V Foot. Real-time processing of social media with sentinel: a syndromic surveillance system incorporating deep learning for health classification.
- [94] Steve Hanneke and Dan Roth. Iterative labeling for semi-supervised learning. Technical report, University of Illinois, Urbana, IL, USA, 2004.
- [95] Nitin Hardeniya. *NLTK essentials*. Packt Publishing Ltd, 2015.
- [96] Nitin Hardeniya. *NLTK essentials*. Packt Publishing Ltd, 2015.

- [97] Wolfgang Karl Härdle, Marlene Müller, Stefan Sperlich, and Axel Werwatz. *Nonparametric and semiparametric models*. Springer Science & Business Media, 2012.
- [98] Jenine K Harris, Raed Mansour, Bechara Choucair, Joe Olson, Cory Nissen, and Jay Bhatt. Health department use of social media to identify foodborne illness—chicago, illinois, 2013–2014. *MMWR. Morbidity and mortality weekly report*, 63(32):681, 2014.
- [99] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [100] N. Heavilin, B. Gerbert, J.E. Page, and J.L. Gibbs. Public health surveillance of dental pain via twitter. *Journal of Dental Research*, 90(9):1047–1051, jul 2011.
- [101] Marti A Hearst. Direction-based text interpretation as an information access refinement. *Text-based intelligent systems: current research and practice in information extraction and retrieval*, pages 257–274, 1992.
- [102] Marti A Hearst. Text data mining: Issues, techniques, and the relationship to information access. In *Presentation notes for UW/MS workshop on data mining*, pages 112–117, 1997.
- [103] Kelly J Henning. What is syndromic surveillance. *Morbidity and mortality weekly report*, 53(Supplement):7–11, 2004.
- [104] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [105] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.
- [106] George Hripcsak and Adam S Rothschild. Agreement, the F-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.
- [107] Yulin Hswen, Qiuyuan Qin, John S. Brownstein, and Jared B. Hawkins. Feasibility of using social media to monitor outdoor air pollution in london, england. *Preventive Medicine*, 121:86–93, apr 2019.
- [108] Hongping Hu, Haiyan Wang, Feng Wang, Daniel Langley, Adrian Avram, and Maoxing Liu. Prediction of influenza-like illness based on the improved artificial tree algorithm and artificial neural network. *Scientific Reports*, 8(1), mar 2018.
- [109] Jiangmiao Huang, Hui Zhao, and Jie Zhang. Detecting flu transmission by social sensor in china. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. IEEE, aug 2013.

- [110] Alison Huettner and Pero Subasic. Fuzzy typing for document management. *ACL 2000 Companion Volume: Tutorial Abstracts and Demonstration Notes*, pages 26–27, 2000.
- [111] Hayate Iso, Shoko Wakamiya, and Eiji Aramaki. Conditional density estimation of tweet location: A feature-dependent approach. In *MED-INFO 2017: Precision Healthcare Through Informatics: Proceedings of the 16th World Congress on Medical and Health Informatics*, volume 245, page 408. IOS Press, 2018.
- [112] Peter Jackson and Isabelle Moulinier. *Natural language processing for online applications: Text retrieval, extraction and categorization*, volume 5. John Benjamins Publishing, 2007.
- [113] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- [114] Anil K Jain and Patrick J Flynn. *Image segmentation using clustering*. IEEE Press, Piscataway, NJ, 1996.
- [115] Daniel Janies, Z Witter, Christian Gibson, Thomas Kraft, Izzet F Senturk, and Ü Çatalyürek. Syndromic surveillance of infectious diseases meets molecular epidemiology in a workflow and phylogeographic application. *Studies in health technology and informatics*, 216:766–770, 2015.
- [116] Sage Jenson, Majerle Reeves, Marcello Tomasini, and Ronaldo Menezes. Mining location information from users' spatio-temporal data. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*. IEEE, aug 2017.
- [117] Lifeng Jin and William Schuler. A comparison of word similarity performance using explanatory and non-explanatory texts. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 990–994, 2015.
- [118] Lifeng Jin and William Schuler. A comparison of word similarity performance using explanatory and non-explanatory texts. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 990–994, 2015.
- [119] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938, 2011.

- [120] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.
- [121] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.
- [122] Prashant D Joshi, MS Bewoor, and SH Patil. In text mining. 1963.
- [123] Jihoon Jung and Christopher K. Uejio. Social media responses to heat waves. *International Journal of Biometeorology*, 61(7):1247–1260, jan 2017.
- [124] Dan Jurafsky. *Speech & language processing*. Pearson Education India, 2000.
- [125] Matjaz Juršic, Igor Mozetic, Tomaz Erjavec, and Nada Lavrac. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9):1190–1214, 2010.
- [126] Ireneus Kagashe, Zhijun Yan, and Imran Suheryani. Enhancing seasonal influenza surveillance: Topic analysis of widely used medicinal drugs using twitter data. *Journal of Medical Internet Research*, 19(9):e315, sep 2017.
- [127] Gloria J. Kang, Sinclair R. Ewing-Nelson, Lauren Mackey, James T. Schlitt, Achla Marathe, Kaja M. Abbas, and Samarth Swarup. Semantic network analysis of vaccine sentiment in online social media. *Vaccine*, 35(29):3621–3638, jun 2017.
- [128] Mark Kantrowitz. Method and apparatus for analyzing affect and emotion in text, September 16 2003. US Patent 6,622,140.
- [129] Fred Karlsson, Atro Voutilainen, Juha Heikkilae, and Arto Anttila. *Constraint Grammar: a language-independent system for parsing unrestricted text*, volume 4. Walter de Gruyter, 1995.
- [130] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.
- [131] James Kennedy. Particle swarm optimization. *Encyclopedia of machine learning*, pages 760–766, 2010.
- [132] Yasmin Khan, Garvin J. Leung, Paul Belanger, Effie Gournis, David L. Buckeridge, Li Liu, Ye Li, and Ian L. Johnson. Comparing twitter data to routine data sources in public health surveillance for the 2015 pan/parapan american games: an ecological study. *Canadian Journal of Public Health*, 109(3):419–426, apr 2018.

- [133] David Khanaferov, Christopher Luc, and Taehyung Wang. Social network data mining using natural language processing and density based clustering. In *2014 IEEE International Conference on Semantic Computing*. IEEE, jun 2014.
- [134] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
- [135] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.
- [136] Benjamin King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(317):86–101, 1967.
- [137] Ioannis Korkontzelos, Azadeh Nikfarjam, Matthew Shardlow, Abeer Sarker, Sophia Ananiadou, and Graciela H. Gonzalez. Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts. *Journal of Biomedical Informatics*, 62:148–158, aug 2016.
- [138] Moritz U. G. Kraemer, D. Bisanzio, R. C. Reiner, R. Zakar, J. B. Hawkins, C. C. Freifeld, D. L. Smith, S. I. Hay, J. S. Brownstein, and T. Alex Perkins. Inferences about spatiotemporal variation in dengue virus transmission are sensitive to assumptions about human mobility: a case study using geolocated tweets from lahore, pakistan. *EPJ Data Science*, 7(1), jun 2018.
- [139] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 61–68. ACM, 2009.
- [140] M. Krieck, L. Otrusina, P. Smrz, P. Dolog, W. Nejdl, E. Velasco, and K. Denecke. How to exploit twitter for public health monitoring? *Methods of Information in Medicine*, 52(04):326–339, 2013.
- [141] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [142] Robert Krovetz. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202. ACM, 1993.
- [143] Arun CS Kumar and Suchendra M. Bhandarkar. A deep learning paradigm for detection of harmful algal blooms. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, mar 2017.

- [144] Emine Ela Küçük, Kürşad Yapar, Dilek Küçük, and Doğan Küçük. Ontology-based automatic identification of public health-related turkish tweets. *Computers in Biology and Medicine*, 83:1–9, apr 2017.
- [145] Alex Lamb, Michael J Paul, and Mark Dredze. Separating fact from fear: Tracking flu infections on twitter. In *HLT-NAACL*, pages 789–795, 2013.
- [146] Vasileios Lamos and Nello Cristianini. Tracking the flu pandemic by monitoring the social web. In *Cognitive Information Processing (CIP), 2010 2nd International Workshop on*, pages 411–416. IEEE, 2010.
- [147] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.
- [148] Gyemin Lee and Clayton Scott. Em algorithms for multivariate gaussian mixture models with truncated and censored data. *Computational Statistics & Data Analysis*, 56(9):2816–2829, 2012.
- [149] Hopin Lee, James H McAuley, Markus Hübscher, Heidi G Allen, Steven J Kamper, and G Lorimer Moseley. Tweeting back: predicting new cases of back pain with mass social media data. *Journal of the American Medical Informatics Association*, 23(3):644–648, dec 2015.
- [150] Kathy Lee, Ankit Agrawal, and Alok Choudhary. Real-time disease surveillance using twitter data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*. ACM Press, 2013.
- [151] Kathy Lee, Ankit Agrawal, and Alok Choudhary. Mining social media streams to improve public health allergy surveillance. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15*. ACM Press, 2015.
- [152] Kathy Lee, Ankit Agrawal, and Alok Choudhary. Forecasting influenza levels using real-time social media streams. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, aug 2017.
- [153] Anton V Leouski and W Bruce Croft. An evaluation of techniques for clustering search results. Technical report, DTIC Document, 2005.
- [154] Sunghoon Lim, Conrad S. Tucker, and Soundar Kumara. An unsupervised machine learning model for discovering latent infectious diseases using social media data. *Journal of Biomedical Informatics*, 66:82–94, feb 2017.

- [155] Dekang Lin and Patrick Pantel. Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM, 2001.
- [156] Guangyu Lin, Razieh Nokhbeh Zaeem, Haowei Sun, and K. Suzanne Barber. Trust filter for disease surveillance: Identity. In *2017 Intelligent Systems Conference (IntelliSys)*. IEEE, sep 2017.
- [157] Wei-San Lin, Hong-Jie Dai, Jitendra Jonnagaddala, Nai-Wun Chang, Toni Rose Jue, Usman Iqbal, Joni Yu-Hsuan Shao, I-Jen Chiang, and Yu-Chuan Li. Utilizing different word representation methods for twitter data in adverse drug reactions extraction. In *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. IEEE, nov 2015.
- [158] Ken Litkowski. Feature ablation for preposition disambiguation. Technical report, CL Research, Damascus, MD, USA, 2016.
- [159] Sam Liu, Miaoqi Zhu, Dong Jin Yu, Alexander Rasin, and Sean D Young. Using real-time social media technologies to monitor levels of perceived stress and emotional state in college students: A web-based questionnaire study. *JMIR Mental Health*, 4(1):e2, jan 2017.
- [160] Evan Dennison Livelo and Charibeth Cheng. Intelligent dengue in-foveillance using gated recurrent neural learning and cross-label frequencies. In *2018 IEEE International Conference on Agents (ICA)*. IEEE, jul 2018.
- [161] Nikola Ljubešić and Darja Fišer. A global analysis of emoji usage. *ACL 2016*, page 82, 2016.
- [162] Julie B Lovins. Development of a stemming algorithm. 1968.
- [163] Julie B Lovins. Development of a stemming algorithm. 1968.
- [164] Fred Sun Lu, Suqin Hou, Kristin Baltrusaitis, Manan Shah, Jure Leskovec, Rok Sasic, Jared Hawkins, John Brownstein, Giuseppe Conidi, Julia Gunn, Josh Gray, Anna Zink, and Mauricio Santillana. Accurate influenza monitoring and forecasting using novel internet data streams: A case study in the boston metropolis. *JMIR Public Health and Surveillance*, 4(1):e4, jan 2018.
- [165] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [166] Tim K. Mackey and Janani Kalyanam. Detection of illicit online sales of fentanyl via twitter. *F1000Research*, 6:1937, nov 2017.

- [167] Tim K. Mackey, Janani Kalyanam, Takeo Katsuki, and Gert Lanckriet. Twitter-based detection of illegal online sale of prescription opioid. *American Journal of Public Health*, 107(12):1910–1915, dec 2017.
- [168] Prasenjit Majumder, Mandar Mitra, Swapan K Parui, Gobinda Kole, Pabitra Mitra, and Kalyankumar Datta. Yass: Yet another suffix stripper. *ACM transactions on information systems (TOIS)*, 25(4):18, 2007.
- [169] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.
- [170] Mihai David Marin. Effectiveness of neural language models for word prediction of textual mammography reports. B.S. thesis, University of Twente, 2019.
- [171] Dan C. Marinescu. Nature-inspired algorithms and systems. In *Complex Systems and Clouds*, pages 33–63. Elsevier, 2017.
- [172] Philip M Massey, Amy Leader, Elad Yom-Tov, Alexandra Budenz, Kara Fisher, and Ann C Klassen. Applying multiple data collection tools to quantify human papillomavirus vaccine communication on twitter. *Journal of medical Internet research*, 18(12), 2016.
- [173] Chandler McClellan, Mir M Ali, Ryan Mutter, Larry Kroutil, and Justin Landwehr. Using social media to monitor mental health discussions - evidence from twitter. *Journal of the American Medical Informatics Association*, page ocw133, oct 2016.
- [174] Karen McCullagh. Blogging: self presentation and privacy. *Information & communications technology law*, 17(1):3–23, 2008.
- [175] Massimo Melucci and Nicola Orio. A novel method for stemmer generation based on hidden markov models. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 131–138. ACM, 2003.
- [176] Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978.
- [177] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [178] David Mimno and Andrew McCallum. Organizing the oca: learning faceted subjects from a library of digital books. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 376–385. ACM, 2007.

-
- [179] Tom M Mitchell. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.
 - [180] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G Altman, Prisma Group, et al. Preferred reporting items for systematic reviews and meta-analyses: the prisma statement., 2010.
 - [181] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 341–349. ACM, 2002.
 - [182] Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M Carley. Is the sample good enough? comparing data from twitter’s streaming api with twitter’s firehose. In *Seventh international AAAI conference on weblogs and social media*, 2013.
 - [183] Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M Carley. Is the sample good enough? comparing data from twitter’s streaming api with twitter’s firehose. In *Seventh international AAAI conference on weblogs and social media*, 2013.
 - [184] Julia Murphy and Max Roser. Internet. *Our World in Data*, 2019. <https://ourworldindata.org/internet>.
 - [185] Eugene W Myers. An o (nd) difference algorithm and its variations. *Algorithmica*, 1(1):251–266, 1986.
 - [186] Priya Nambisan, Zhihui Luo, Akshat Kapoor, Timothy B. Patrick, and Ron A. Cisler. Social media, big data, and public health informatics: Ruminating behavior of depression revealed through twitter. In *2015 48th Hawaii International Conference on System Sciences*. IEEE, jan 2015.
 - [187] Kruti Nargund and S. Natarajan. Public health allergy surveillance using micro-blogs. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, sep 2016.
 - [188] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM, 2003.
 - [189] Brad L Neiger, Rosemary Thackeray, Scott H Burton, Christophe G Giraud-Carrier, and Michael C Fagen. Evaluating social media’s capacity to develop engaged audiences in health promotion settings: use of twitter metrics as a case study. *Health promotion practice*, 14(2):157–162, 2013.

- [190] Dean Neu, Greg Saxton, Abu Rahaman, and Jeffery Everett. Twitter and social accountability: Reactions to the panama papers. *Critical Perspectives on Accounting*, 2019.
- [191] David Newman, Kat Hagedorn, Chaitanya Chemudugunta, and Padhraic Smyth. Subject metadata enrichment using statistical topic models. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 366–375. ACM, 2007.
- [192] Kyosuke Nishida, Ryohei Banno, Ko Fujimura, and Takahide Hoshide. Tweet classification by data compression. In *Proceedings of the 2011 international workshop on DETecting and Exploiting Cultural diversity on the social web*, pages 29–34. ACM, 2011.
- [193] Jakub Nowak, Ahmet Taspinar, and Rafał Scherer. Lstm recurrent neural networks for short text and sentiment classification. In *International Conference on Artificial Intelligence and Soft Computing*, pages 553–562. Springer, 2017.
- [194] Rebecca Nugent and Marina Meila. An overview of clustering applied to molecular biology. *Statistical methods in molecular biology*, pages 369–404, 2010.
- [195] Bahadorreza Ofoghi, Meghan Mann, and Karin Verspoor. Towards early discovery of salient health threats: A social media emotion classification technique. In *Biocomputing 2016: Proceedings of the Pacific Symposium*, pages 504–515. World Scientific, 2016.
- [196] Karen O’Connor, Pranoti Pimpalkhute, Azadeh Nikfarjam, Rachel Ginn, Karen L Smith, and Graciela Gonzalez. Pharmacovigilance on twitter? mining tweets for adverse drug reactions. In *AMIA annual symposium proceedings*, volume 2014, page 924. American Medical Informatics Association, 2014.
- [197] David D Palmer and Marti A Hearst. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23(2):241–267, 1997.
- [198] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics, 2005.
- [199] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

- [200] Jon Parker, Yifang Wei, Andrew Yates, Ophir Frieder, and Nazli Goharian. A framework for detecting public health trends with twitter. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM '13*. ACM Press, 2013.
- [201] Jon Parker, Andrew Yates, Nazli Goharian, and Ophir Frieder. Health-related hypothesis generation using social media data. *Social Network Analysis and Mining*, 5(1), mar 2015.
- [202] Matthias Benjamin Passer. Verb classifiers- misfits of nominal classification? In *35th TABU-dag*, June 2014.
- [203] Fuchun Peng and Dale Schuurmans. Combining naive bayes and n-gram language models for text classification. In *European Conference on Information Retrieval*, pages 335–350. Springer, 2003.
- [204] Yang Peng, Melody Moh, and Teng-Sheng Moh. Efficient adverse drug event extraction using twitter sentiment analysis. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, aug 2016.
- [205] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [206] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [207] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [208] Philipp Petrenz. *Assessing approaches to genre classification*. PhD thesis, M. Sc. thesis, School of Informatics, University of Edinburgh, 2009.
- [209] Mai T Pham, Andrijana Rajić, Judy D Greig, Jan M Sargeant, Andrew Papadopoulos, and Scott A McEwen. A scoping review of scoping reviews: advancing the approach and enhancing the consistency. *Research synthesis methods*, 5(4):371–385, 2014.
- [210] Nhathai Phan, Soon Ae Chun, Manasi Bhole, and James Geller. Enabling real-time drug abuse detection in tweets. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, apr 2017.

- [211] David Pierce and Claire Cardie. Limitations of co-training for natural language learning from large datasets. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 1–9, 2001.
- [212] Pietro Pinoli, Davide Chicco, and Marco Masseroli. Latent dirichlet allocation based on gibbs sampling for gene function prediction. In *Computational Intelligence in Bioinformatics and Computational Biology, 2014 IEEE Conference on*, pages 1–8. IEEE, 2014.
- [213] Stephen Pollock. A rule-based message filtering system. *ACM Transactions on Information Systems (TOIS)*, 6(3):232–254, 1988.
- [214] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [215] Martin F Porter. Snowball: A language for stemming algorithms, 2001.
- [216] Sudha Ram, Wenli Zhang, Max Williams, and Yolande Pengetnze. Predicting asthma-related emergency department visits using big data. *IEEE Journal of Biomedical and Health Informatics*, 19(4):1216–1223, jul 2015.
- [217] Edie M Rasmussen. Clustering algorithms. *Information retrieval: data structures & algorithms*, 419:442, 1992.
- [218] Lisa F Rau. Extracting company names from text. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, volume 1, pages 29–32. IEEE, 1991.
- [219] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. Sentence boundary detection: A long solved problem? *COLING (Posters)*, 12:985–994, 2012.
- [220] Marina Riga and Kostas Karatzas. Investigating the relationship between social media content and real-time observations for urban air quality and public health. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14) - WIMS '14*. ACM Press, 2014.
- [221] Michael D Riley. Some applications of tree-based modelling to speech and language. In *Proceedings of the workshop on Speech and Natural Language*, pages 339–352. Association for Computational Linguistics, 1989.
- [222] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005.

- [223] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, 1998.
- [224] Koustav Rudra, Ashish Sharma, Niloy Ganguly, and Muhammad Imran. Classifying information from microblogs during epidemics. In *Proceedings of the 2017 International Conference on Digital Health - DH '17*. ACM Press, 2017.
- [225] Adam Sadilek, Henry Kautz, Lauren DiPrete, Brian Labus, Eric Portman, Jack Teitel, and Vincent Silenzio. Deploying nemesis: Preventing foodborne illness by data mining social media. *Ai Magazine*, 38(1):37–48, 2017.
- [226] Adam Sadilek, Henry Kautz, and Vincent Silenzio. Predicting disease transmission from geo-tagged micro-blog data. In *AAAI Conference on Artificial Intelligence*, 2012.
- [227] Sumit Saha. A comprehensive guide to convolutional neural networks - the eli5 way. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd6e1452701>
- [228] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
- [229] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.
- [230] Hamman Samuel, Benyamin Noori, Sara Farazi, and Osmar Zaiane. Context prediction in the social web using applied machine learning: A study of canadian tweeters. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, dec 2018.
- [231] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
- [232] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.
- [233] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.
- [234] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.
- [235] Satoshi Sekine et al. Nyu: Description of the japanese ne system used for met-2. In *Proc. Message Understanding Conference*, 1998.

- [236] Ovidiu Serban, Nicholas Thapen, Brendan Maginnis, Chris Hankin, and Virginia Foot. Real-time processing of social media with sentinel: A syndromic surveillance system incorporating deep learning for health classification. *Information Processing & Management*, 56(3):1166–1184, 2019.
- [237] E Severi, E Heinsbroek, C Watson, M Catchpole, et al. Infectious disease surveillance for the london 2012 olympic and paralympic games. *Eurosurveillance*, 17(31):20232, 2012.
- [238] J Danielle Sharpe, Richard S Hopkins, Robert L Cook, and Catherine W Striley. Evaluating google, twitter, and wikipedia as tools for influenza surveillance using bayesian change point analysis: a comparative analysis. *JMIR public health and surveillance*, 2(2), 2016.
- [239] Adrian BR Shatte, Delyse M Hutchinson, and Samantha J Teague. Machine learning in mental health: a scoping review of methods and applications. *Psychological medicine*, pages 1–23, 2019.
- [240] Christopher C Shilakes and Julie Tylman. Enterprise information portals. *Merrill Lynch*, November, 16, 1998.
- [241] Sumit Sidana, Sihem Amer-Yahia, Marianne Clausel, Majdeddine Rebai, Son T. Mai, and Massih-Reza Amini. Health monitoring on social media over time. *IEEE Transactions on Knowledge and Data Engineering*, 30(8):1467–1480, aug 2018.
- [242] Alessio Signorini, Alberto Maria Segre, and Philip M. Polgreen. The use of twitter to track levels of disease activity and public concern in the u.s. during the influenza a h1n1 pandemic. *PLoS ONE*, 6(5):e19467, may 2011.
- [243] Lauren Sinnenberg, Alison M Buttenheim, Kevin Padrez, Christina Mancheno, Lyle Ungar, and Raina M Merchant. Twitter as a tool for health research: a systematic review. *American journal of public health*, 107(1):e1–e8, 2017.
- [244] Alex J Smola et al. Regression estimation with support vector learning machines. *Master’s thesis, Technische Universit at M unchen*, 1996.
- [245] Peter HA Sneath, Robert R Sokal, et al. *Numerical taxonomy. The principles and practice of numerical classification*. 1973.
- [246] Benjamin Snyder and Regina Barzilay. Multiple aspect ranking using the good grief algorithm. In *HLT-NAACL*, pages 300–307, 2007.
- [247] Susie Song and Zina Ben Miled. Digital immunization surveillance: Monitoring flu vaccination rates using online social networks. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, oct 2017.

- [248] William M Spears, Kenneth A De Jong, Thomas Bäck, David B Fogel, and Hugo De Garis. An overview of evolutionary computation. In *European Conference on Machine Learning*, pages 442–459. Springer, 1993.
- [249] Efstathios Stamatatos, Nikos Fakotakis, and George Kokkinakis. Automatic extraction of rules for sentence boundary disambiguation. In *Proceedings of the Workshop on Machine Learning in Human Language Technology*, pages 88–92, 1999.
- [250] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.
- [251] Kun Su, Yu Xiong, Li Qi, Yu Xia, Baisong Li, Lin Yang, Qin Li, Wenge Tang, Xian Li, Xiaowen Ruan, Shaofeng Lu, Xianxian Chen, Chaobo Shen, Jiaying Xu, Liang Xu, Mei Han, and Jing Xiao. City-wide influenza forecasting based on multi-source data. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, dec 2018.
- [252] Lu Tang, Bijie Bie, and Degui Zhi. Tweeting about measles during stages of an outbreak: A semantic network approach to the framing of an emerging infectious disease. *American Journal of Infection Control*, 46(12):1375–1380, dec 2018.
- [253] Narendran Thangarajan, Nella Green, Amarnath Gupta, Susan Little, and Nadir Weibel. Analyzing social media to characterize local HIV at-risk populations. In *Proceedings of the conference on Wireless Health - WH '15*. ACM Press, 2015.
- [254] Nicholas Thapen, Donal Simmie, Chris Hankin, and Joseph Gillard. DEFENDER: Detecting and forecasting epidemics using novel data-analytics for enhanced response. *PLOS ONE*, 11(5):e0155417, may 2016.
- [255] S Triple. Assessment of syndromic surveillance in europe. *Lancet (London, England)*, 378(9806):1833, 2011.
- [256] S Triple. Assessment of syndromic surveillance in europe. *Lancet (London, England)*, 378(9806):1833, 2011.
- [257] Peter D Turney. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655, 2008.
- [258] Peter D Turney and Michael L Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346, 2003.

- [259] Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.
- [260] Prabha Umapathy, C. Venkataseshaiyah, and M. Senthil Arumugam. Particle swarm optimization with various inertia weight variants for optimal power flow solution. *Discrete Dynamics in Nature and Society*, 2010:1–15, 2010.
- [261] Utrecht University Utrecht Institute of Linguistics. Lexicon of linguistics, 1996.
- [262] P Amrutha Valli, M Uma, and T Sasikala. Tracing out various diseases by analyzing twitter data applying data mining techniques. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE, aug 2017.
- [263] Francisco J. Valverde-Albacete and Carmen Peláez-Moreno. 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox. *PLoS ONE*, 9(1):1–10, 01 2014.
- [264] Moritz Wagner, Vasileios Lamos, Ingemar J. Cox, and Richard Pebody. The added value of online user-generated content in traditional methods for influenza surveillance. *Scientific Reports*, 8(1), sep 2018.
- [265] Shoko Wakamiya, Yukiko Kawai, and Eiji Aramaki. Twitter-based influenza detection after flu peak via tweets with indirect information: Text mining study. *JMIR Public Health and Surveillance*, 4(3):e65, sep 2018.
- [266] Feng Wang, Haiyan Wang, Kuai Xu, Ross Raymond, Jaime Chon, Shaun Fuller, and Anton Debruyne. Regional level influenza study with geo-tagged twitter data. *Journal of Medical Systems*, 40(8), jul 2016.
- [267] Junhui Wang, Xiaotong Shen, and Wei Pan. On transductive support vector machines. *Contemporary Mathematics*, 443:7–20, 2007.
- [268] Junxiang Wang, Liang Zhao, and Yanfang Ye. Semi-supervised multi-instance interpretable models for flu shot adverse event detection. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, dec 2018.
- [269] Junxiang Wang, Liang Zhao, Yanfang Ye, and Yuji Zhang. Adverse event detection by integrating twitter data and VAERS. *Journal of Biomedical Semantics*, 9(1), jun 2018.
- [270] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, pages 6677–6686, 2019.

-
- [271] M Wargon, B Guidet, TD Hoang, and G Hejblum. A systematic review of models for forecasting the number of emergency department visits. *Emergency Medicine Journal*, 26(6):395–399, 2009.
- [272] Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision rnns for language recognition. *arXiv preprint arXiv:1805.04908*, 2018.
- [273] Joshua Heber West, Parley Cougar Hall, Carl Lee Hanson, Kyle Prier, Christophe Giraud-Carrier, E Shannon Neeley, and Michael Dean Barnes. Temporal variability of problem drinking on twitter. *Open Journal of Preventive Medicine*, 2(01):43, 2012.
- [274] Janyce Wiebe, Eric Breck, Chris Buckley, Claire Cardie, Paul Davis, Bruce Fraser, Diane J Litman, David R Pierce, Ellen Riloff, Theresa Wilson, et al. Recognizing and organizing opinions expressed in the world press. In *New Directions in Question Answering*, pages 12–19, 2003.
- [275] Janyce M Wiebe and William J Rapaport. A computational theory of perspective and reference in narrative. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, pages 131–138. Association for Computational Linguistics, 1988.
- [276] Wikipedia. How to spot a twitter spambot. <http://mashable.com/2013/11/08/twitter-spambots/#x0RY3kS2ssqP>, 2003.
- [277] C-EA Winslow. The untilled fields of public health. *Science*, pages 23–33, 1920.
- [278] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
- [279] Hyekyung Woo, Hyeon Sung Cho, Eunyoung Shim, Jong Koo Lee, Kihwang Lee, Gilyoung Song, and Youngtae Cho. Identification of keywords from twitter and web blog posts to detect influenza epidemics in korea. *Disaster Medicine and Public Health Preparedness*, 12(03):352–359, jul 2017.
- [280] World Health Organisation WHO. The world health report 2007 - a safer future: global public health security in the 21st century. Available at: <http://www.who.int/whr/2007/en/>, 2007.
- [281] World Health Organisation WHO. The world health report 2007 - a safer future: global public health security in the 21st century. Available at: <http://www.who.int/whr/2007/en/>, 2007.
- [282] Jinxi Xu and W Bruce Croft. Corpus-based stemming using co-occurrence of word.

- [283] Christopher C Yang, Hsinchun Chen, and Kay Hong. Visualization of large category map for internet browsing. *Decision support systems*, 35(1):89–102, 2003.
- [284] Hui Yang, Jamie Callan, and Luo Si. Knowledge transfer and opinion detection in the trec 2006 blog track. In *TREC*, 2006.
- [285] Wei Yang and Lan Mu. GIS analysis of depression among twitter users. *Applied Geography*, 60:217–223, jun 2015.
- [286] Wei Yang, Lan Mu, and Ye Shen. Effect of climate and seasonality on depressed mood among twitter users. *Applied Geography*, 63:184–191, sep 2015.
- [287] Surender Reddy Yerva, Zoltán Miklós, and Karl Aberer. What have fruits to do with technology?: the case of orange, blackberry and apple. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, page 48. ACM, 2011.
- [288] Elad Yom-Tov, Diana Borsa, Ingemar J Cox, and Rachel A McKendry. Detecting disease outbreaks in mass gatherings using internet data. *Journal of Medical Internet Research*, 16(6):e154, jun 2014.
- [289] Sean D. Young, Neil Mercer, Robert E. Weiss, Elizabeth A. Torrone, and Sevgi O. Aral. Using social media as a tool to predict syphilis. *Preventive Medicine*, 109:58–61, apr 2018.
- [290] Alan L Yuille and Anand Rangarajan. The concave-convex procedure. *Neural computation*, 15(4):915–936, 2003.
- [291] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 46–54. ACM, 1998.
- [292] Thomas Zeugmann, Pascal Poupart, James Kennedy, Xin Jin, Jiawei Han, Lorenza Saitta, Michele Sebag, Jan Peters, J. Andrew Bagnell, Walter Daelemans, Geoffrey I. Webb, Kai Ming Ting, Kai Ming Ting, Geoffrey I. Webb, Jelber Sayyad Shirabad, Johannes Fürnkranz, Eyke Hüllermeier, Stan Matwin, Yasubumi Sakakibara, Pierre Flener, Ute Schmid, Cecilia M. Procopiuc, Nicolas Lachiche, and Johannes Fürnkranz. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer US, 2011.
- [293] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32. ACM, 2003.

-
- [294] Liang Zhao, Jiangzhuo Chen, Feng Chen, Wei Wang, Chang-Tien Lu, and Naren Ramakrishnan. Simnest: Social media nested epidemic simulation via online semi-supervised deep learning. In *2015 IEEE International Conference on Data Mining*, pages 639–648. IEEE, 2015.
 - [295] Bin Zou, Vasileios Lamos, Russell Gorton, and Ingemar J. Cox. On infectious intestinal disease surveillance using social media content. In *Proceedings of the 6th International Conference on Digital Health Conference - DH '16*. ACM Press, 2016.
 - [296] Ovidiu Șerban, Nicholas Thapen, Brendan Maginnis, Chris Hankin, and Virginia Foot. Real-time processing of social media with SENTINEL: A syndromic surveillance system incorporating deep learning for health classification. *Information Processing & Management*, 56(3):1166–1184, may 2019.

A:

Twitter Data Collection

Keywords

pollution, smog, poor air quality, wheeze, wheezing, difficulty breathing, asthma, inhaler, air pollution, itchy eyes, sore eyes, trouble breathing, cannot breathe, could not breathe, can't breathe, couldn't breathe, asma, short of breath, tight chest, chest tightness, respiratory disease, pea souper, murk, fumes, acid rain, gasping, puffing, panting.

B:

Positive Word Dictionary

adore, adorable, accomplish, achievement, achieve, action, active, admire, adventure, agree, agreeable, amaze, amazing, angel, approve, attractive, awesome, beautiful, brilliant, bubbly, calm, celebrate, celebrating, charming, cheery, cheer, clean, congratulation, cool, cute, divine, earnest, easy, ecstasy, ecstatic, effective, effective, efficient, effortless, elegant, enchanting, encouraging, energetic, energized, enthusiastic, enthusiasm, excellent, exciting, excited, fabulous, fair, familiar, famous, fantastic, fine, fit, fortunate, free, fresh, friend, fun, generous, genius, glowing, good, great, grin, handsome, happy, hilarious, hilarity, lmao, lol, rofl, haha, healthy, ideal, impressive, independent, intellectual, intelligent, inventive, joy, keen, laugh, legendary, light, lively, lovely, lucky, marvel, nice, okay, paradise, perfect, pleasant, popular, positive, powerful, pretty, progress, proud, quality, refresh, restore, right, smile, success, sunny, super, wealthy, money, cash, well, wonderful, wow, yes, yum.

C:

Negative Word Dictionary

abysmal, adverse, alarming, angry, rage, annoy, anxious, anxiety, attack, appalling, atrocious, awful, bad, broken, can't, not, cant, cannot, cold, collapse, crazy, cruel, cry, damage, damaging, depressed, depression, dirty, disease, disgust, distress, don't, dont, dreading, dreadful, dreary, fail, fear, scare, feeble, foul, fright, ghastly, grave, greed, grim, gross, grotesque, gruesome, guilty, hard, harm, hate, hideous, horrible, hostile, hurt, icky, ill, impossible, injure, injury, jealous, lose, lousy, messy, nasty, negative, never, no, nonsense, crap, shit, fuck, fukk, fuxk, nausea, nauseous, pain, reject, repulsive, repulse, revenge, revolting, rotten, rude, ruthless, sad, scary, severe, sick, slimy, smelly, sorry, sticky, stinky, stormy, stress, stuck, stupid, tense, terrible, terrifying, threaten, ugly, unfair, unhappy, unhealthy, unjust, unlucky, unpleasant, upset, unwanted, unwelcome, vile, wary, weary, wicked, worthless, wound, yell, yucky

D:

PSO-Based Keyword Selection

Candidate Set

pollution, smog, poor air quality, wheeze, wheezing, difficulty breathing, asthma, inhaler, air pollution, itchy eyes, sore eyes, trouble breathing, cannot breathe, could not breathe, can't breathe, couldn't breathe, asma, short of breath, tight chest, chest tightness, respiratory disease, pea souper, murk, fumes, acid rain, gasping, puffing, panting, breath, breathing, cant, cannot, crying, omfg, murked, nigha, knocc, bodied, snuffed, cuhz, allergy, allergies, bronchitis, disease, inhaler, symptoms, groaning, whimpering, purring, sweating, shivering, sighing, fume, critiques, tomes, tires, bois, insultes, neck, shoulders, stomach, arms, shoulder, arm, hyperventilating, coughing, sniffing, whimpering, coughin, chokinggasps, facepalm, headdesk, faints, chuckle, groan, puffin, huffing, passin, blowin, belching, puffed, asthma, vicks, earpiece, ventolin, tissues.